# RSTS 50th Birthday Celebration

RSTS-11 was "born" on 11th June 1970 as shown when you PRINT DATE$(1%) with Star Date format selected.

So, this amazing operating system is now 50 years old.

The following 50 memories are from people who developed, enhanced, sold, or used RSTS-11, RSTS/E or CTS500 at some point in their lives.

As Editor, I have taken liberties with the word "RSTS", combining RSTS-11 with RSTS/E, allowing us to celebrate both as 50 years old:

> *RSTS/E has a bunch of new stuff in it to deal with mapping but the bulk of the code carries over from RSTS-11. For example, the file system code is basically the same, as is a large fraction of driver code as well as core kernel services.*
>
> *RSTS/E did lose some oddball things, such as the fact that RSTS-11 did Basic-Plus string garbage collection through a file processor overlay*.

This document contains some amazing examples of how the world has changed during the last 50 years: here are some examples of DEC in the 1970s Vs Apple in 2020:

> a 40 megabyte DEC disk drive was the size of a washing machine while in 2020 you get 32 gigabytes of memory with an Apple Watch
>
> on average, there were about 20 people in RSTS Development while Apple has 800 people just to improve the camera
>
> RSTS and Basic-Plus were restricted to 32K byte maximum program size while iTunes needs 320 megabytes of memory to run on a PC.

A "thank you" to everyone who contributed. And a special "thank you" to all the members of the RSTS Development Team who created a product that wasn't just used by the Users, it was loved by the Users.

Please note there are two appendices at the end, starting at page 84:

> ?Program lost-Sorry
> RSTS In Peace

The Editor can be reached by email: 50years@silverware.co.uk

This is Version 50.0.7 dated 20-Jul-20

RSTS Release Dates. Thanks to http://gunkies.org/wiki/RSTS/E and https://en.wikipedia.org/wiki/RSTS/E

## RSTS-11

| | | |
|---|---|---|
| 1 | Jun-70 | Never released |
| 2A-19 | 1971 | Installed at Carleton College and Seattle Pacific College |
| 2B | Jun-71 | Installed at the Delaware School Auxiliary Association |
| 2C | Oct-71 | Enhanced software support (record I/O) |
| 3A-19 | Jan-72 | |
| 3B | May-72 | |
| 3C | Jun-72 | Enhanced software support (UPDATE mode) |
| 4A-12 | Oct-72 | New hardware supported. |
| 4B | Jul-75 | A patched version of V4A with no new features; last version for non-MMU machines |

## RSTS/E

| | | |
|---|---|---|
| 5A-21 | Jul-73 | Memory Management Unit (MMU) extensions from 64KB to 256KB |
| 5B-24 | Nov-74 | Multiple SWAP files, RTS and CCL introduced |
| 5C-01 | Mar-75 | A patched version of V5B with no new features |
| 6A-02 | Aug-75 | New hardware, 22 bit addressing (2 MB), 63 jobs. Last version installed using DOS-11 |
| 6B-02 | Feb-77 | New hardware, SYSGEN under RT-11, CCL at runtime |
| 6C-03 | Feb-78 | DECnet Phase II, DTR, DIBOL supported. SUPERVISOR mode |
| 7.0 | Aug-78 | New hardware, (4 MB), RDC support |
| 7.1 | Feb-81 | Kernel mode Instruction & Data space support, DECnet Phase III |
| 7.2 | Aug-82 | New hardware (UDA50 drives) |
| 8.0-06 | Apr-83 | New hardware (MicroPDP-11) |
| 9.0-14 | May-85 | New hardware, DCL default, PBS, multiple privs, hashed passwords, new Backup utility,.. |
| 9.1-05 | Oct-85 | Enhanced hardware support, multi-threaded FIP, LOAD INDEX |
| 9.2-10 | Jun-86 | New hardware supported |
| 9.3-20 | Jan-87 | Enhanced hardware support, DECnet Phase IV |
| 9.4 | Jul-87 | New hardware, Scheduler and Terminal Service improvements |
| 9.5-08 | Dec-87 | Clustersize increased to 64, BACKUP/DIR added |
| 9.6 | Sep-88 | LAT support, HELP SPIKE |
| 9.7 | Jul-89 | New hardware, DV0: at runtime, ANSI PDP-11C available |
| 10 | May-90 | Command line history in TTDRV |
| 10.1 | Sep-92 | |
| 10.1A | 1998 | First and last Mentec release that fixes Y2K |

## Adie Old, Applications programmer, Jackson Shipley

Late August 1987 was a special time for me
A new job as a Programmer at Jackson Shipley
First day memories of a room filled with a PDP
Oh and a bloody big book to read, entitled RSTS/E

Resource Sharing and Time Sharing and other wizardry
Basic-Plus as the language, as if it were made for me
Friendly Hellos and Goodbyes on the screens to see
Oh, I was going to get on with this impressive RSTS/E

Basic-Plus wasn't prestigious but it was THE language for me
Developing programmes brought out my artistry
It grew and evolved with BAS24K and BP3
Enhancing the experience of our beloved RSTS/E

The hardware was immense and they called it a PDP
All drawers, cables and boards covered in circuitry
I loved the hum and buzz of this electric monstrosity
Purposely built to look after our RSTS/E

I remember "portable" disk drives, Bloody hell, they were heavy
The LA180 machine, detailing the progress history
VT100s and green screens for all to see
All the tools required to showcase the impressive RSTS/E

But, things weren't plain sailing, as I was about to see
DEC engineers became like friends, drinking all our coffee
Many nights of no sleep were spent repairing the PDP
All of this necessary to look after RSTS/E

And so, 50 years have passed but for me only 33
I still develop in Basic, it's like my right arm you see
Looking back on many memories and lots of history
It just remains for me to say, Happy Birthday and BYE/P

## Adrian Lumsden, RSX Guru

Good luck with your RSTS 50th.  I think that even if I stretch things a little RSX was first released in 1971 (as RSX-15 for the PDP-15) so RSTS was earlier.

I did do a little work on RSTS but it was very little.

I got dragged in to do some consulting on a system that wouldn't work properly and had been told that it was RSX. It turned out to be RSTS.

I did manage to get it to work and without having to resort to migrating them to RSX. I think that I also did a couple of migration jobs from RSTS to RSX. I don't remember why they wanted to do that and it wasn't at my urging.

*Editor's note: Adrian undertook the typesetting of the RSTS 80th Birthday docs.*

## Alan Pickwick, Manchester Grammar School

My first use of an interactive computer system was the new PDP-11/04 that Manchester Grammar School bought from Systime in 1977. With building modifications, about £30,000.

We had four VDUs, a DECWriter, a CDC 400 lpm lineprinter, a 600 ft magnetic tape drive, a card reader (never used!) and two 20 Mbyte exchangeable disk drives, similar to the DEC RPR02.

My first useful program was to store the names and address of my local musical theatre group. When that worked, I modified it to store the names and addresses of the boys of the school. It used Record I/O.

It was our first automation (3-up sticky address labels), the best thing since sliced bread!!! The data has gone through about six different computers since and we now store about 11,000 current and live alumni records but it all started with RSTS/E.

We used RSTS/E to introduce a whole generation of pupils to programming.

And I actually got a 9600 bps serial link to work at a distance of 600 feet!!!

When we scrapped the PDP-11/34 as it was later, I preserved some program code. Here is one statement.

```
30    CCL=1%                                                  &
      FIL$=SYS(CHR$(7%))                                      &
      Z%=INSTR(1%,FIL$," ")                                   &
      FIL$=RIGHT(FIL$,Z%+1%)                                  &
      Z$=SYS(CHR$(8%))                                        &
      IF Z%=0% OR FIL$=="" THEN CCL%=0%                       &
```

And I still think of a RSTS/E error message – something to do with opening files – "Wrong Mode" - when I make an incorrect choice in life.

Regards, Alan Pickwick, Physics Teacher MGS Retired
Honorary Secretary of the North West Computer Conservation Group of the British Computer Society

Alan_C_Pickwick@btinternet.com

# Amanda Spencer, 12 years at Silver Programs

Towards the end of 1988, I applied for a job advertised in my local rag. I was rejected (very politely) at round one. Hey ho I thought, shame.

Some weeks later the phone rang while I was having a bath. You remember those devices attached to the wall, sitting in a cradle attached with a curly-wurly cord not possible (and probably dangerous) to use in the bath, the mobile a figment in some sci-fi imagination. Anyway my significant other Adrian (now husband) told the caller I was in the bath and could he call back some while later. Which he did.

Opening words *"Are you clean?"* Followed by something like *"Would you like to come for a cup of coffee, I may have been a bit hasty in rejecting you for my office vacancy, you may not have a clue what is a PDP-11, but you've got "O"-level Latin...*

Poor Adrian was severely chastised for telling complete strangers that I was in the bath, especially men that turn out to be potential employers.

However, that cup of coffee led to a 12-year long Resource Sharing Time Sharing professional relationship that has been /Extended to a lifelong and enriching friendship.

Together we have been through a pile of success and adversity, shared laughter and tears, mountains were climbed and skied down, oceans crossed and dived into, I learnt what is a PDP-11 and what is a 911, there's so much more with so many memories I don't know where to start.

## Amanda's less formal memories…

Like when the NatWest Bank said they wanted to visit at Abbots Dwell (a 200 year old thatched cottage, the opposite of high tech) and we were like *"Holy Crap Batman, our cover is blown".* We made a welcoming fridge magnet message on the white board and after an afternoon drinking white wine in the garden, they left the PO on your desk.

Or the times in the heat wave when we set up the phones in the garden to be able to sunbathe while selling or doing mail-shot envelope stuffing and people saying *"Can I hear birds?*

Or you put a customer on hold with the words, *"Could you hang on a minute? My colleague just needs some assistance."* whereupon you helped Turbo get his nose and ice-cream pot un-stuck from under a desk.

Or at Tim's we were going to be inspected by Enskilda and he hung jackets on every chair saying his other colleagues had just popped out for a bit.

And the Billy Maxman FedEx Christmas cards episode…

## Archie Reid, Director Hill Samuel

RSTS and PDP-11/70s proved to be the ideal choice of OS and hardware platform on which to base our distributed processing strategy in the mid-1970s, a period in which integrated, centralised mainframe solutions were the norm in banking applications. It was recognised that the business needs in wholesale banking required a systems architecture which was responsive to frequent changes in business strategies and market practice. This was reflected in our modular design of each business function and of individual applications and their underlying processes.

RSTS enabled us to extend this functional modularity to the programming level which proved to have further significant benefits. The program size limitation of 16k words and limited memory addressability imposed the need for highly efficient programming standards and disciplines which prevailed throughout the 1980s after these limitations had been removed in later versions of RSTS. This period saw substantial changes in business needs and market practice – including 'Big Bang' – and our program structures contributed greatly to ease of maintenance and functional enhancement.

RSTS was therefore central to our achievement of creating a systems and IT architecture which was responsive to changing business needs as well being a highly cost-effective solution. In the late 1980s we came under pressure from DEC to redevelop our operational systems on the VAX range which we had adopted for centralised systems such as our marketing database and management information. However, we decided that our RSTS based systems could have a lot more value to offer into the 1990s. It had become evident that cost/performance trends in IT hardware were improving year on year and we anticipated that a further level of distributed processing to the desktop would eventually be economically viable. In this scenario our RSTS based program structures could yield yet another substantial return in investment.

## Brett Bump, www.rsts.org

This is the home of the RSTS.org web site. RSTS is an operating system that was created by Digital Equipment Corporation to run on the 16-bit PDP-11 series of computers. For more information about RSTS, you should check out the RSTS/E article about it on Wikipedia.

RSTS.org contains a repository of RSTS distros, information about RSTS and this blog that people can use to discuss the system that got most of us started.

Among the gems on this web site, you will find…

| | |
|---|---|
| Emulators | http://www.rsts.org/autoindex.php?dir=emulators |
| RSTS Professional | http://www.rsts.org/autoindex.php?dir=rstspro |
| S/w distribution | http://www.rsts.org/autoindex.php?dir=distros |

*Editor's note: Brett has been maintaining his amazing rsts.org website for a very long time. Sadly I have failed to get in contact with him: our last exchange of emails was nine years ago…*

## David Allen, Allen Computers

Soooo much stuff still in my head as if it were yesterday.

Am I being picky when I say that it was actually RSTS that was born on that date so RSTS/E will be fifty years old a little later?

I was thinking that the first version of RSTS/E came along in 1973…?

I remember starting work on RSTS V4A-12 in 1972 at the DEC offices in Fountain House and Arkwright Road in Reading while we were waiting for our own PDP-11/40 to arrive the following year in 1973. Oh my… yes… just remembered storing my code on DECtapes in between visits… a random access tape drive… wow!

That was a scary year. I was a tender 21 years old and had this huge and scary order on its way.

Remembering the system that was put into Green Shield (later spawning Argos) in Daventry. Shed loads of early problems as they had the RKII-D controller for their RK05 drives. We were lucky as our system had come in somewhat earlier and used the RK11-C controller. The RK11-D was causing random crashes for an unknown reason. I remember the system being surrounded in wire netting at one stage to create a Faraday cage as it was thought the nearby powerful radio transmitter was a possible cause. What was it in the end? Was a dry joint on the board?... or a problem with the traces being too close? Can't remember!

… but that led to meeting Eoin Hanley. I think he was with CAP and they were writing the software? It also led to meeting the wonderful Sean O'Callaghan from Green Shield.

… and the amazing Alex Taylor from Digital's engineering side… he solved some issues over the years. Truly dedicated.

… And Kevin Mitnick (who I followed but never met), one of the world's most famous early hackers… one of his first famous hacks was into DEC's RSTS/E development system. This wasn't the method he used but I am remembering all those systems that had System Manager [1,1] with a password of SYSMAN and System Library [1,2] with a password of SYSLIB. It was a fun time to be a hacker and there were no IP addresses to track you and BT's Strowger exchanges enabled untraceable calls… except in those days BT didn't exist – it was Post Office Telecommunications… and then there was all the fun with DECnet…

Oh my… and meeting Hamish Donaldson (quite my hero!) when he was still with Hill Samuel… and Archie Reid…

By the way… Wondering if you had seen this for those who miss the old days! A replica PDP-11/70!

https://obsolescence.wixsite.com/obsolescence/pidp-11

I think we set the switches to load address 17765744 to boot it? Why are strange numbers still in our heads after such a long time?

## David Peel, British Steel, Quantic Computing

I think I was the first UK non-DEC employee to use a RSTS system. I logged on to a system called "EUROPE" in Reading at 02:54 PM on 9th January 1972. My first program was the immediate mode command "PRINT 12^2". The system displayed "144".

At the end of a three-hour session, I had written a slightly larger program to generate lots of Pythagorean triples. For my managerial peace of mind, I checked that the program could be saved to DECtape and restored from a different handler at a different terminal into a different account.

Apparently while the DEC "minder" thought I was mad, she had never worked on other computer systems where nothing was taken for granted.

I came away impressed… and British Steel placed an order.

*Editor's note: Sadly David died in February 2017.*

# Dennis Hayden, DEC Internal Systems 1970s

I was the manager of the internal DEC OAS (Order Administration System), based in Reading, that ran on a RSTS/E system. OAS was also used in every country in EMEA (Europe, Middle East and Africa), a total of 17 countries, with one RSTS system per country.

We had a mass of reports, written by a Porsche driving contractor, including weekly programs that analysed orders by Product Line, Product Group, Salesperson (we called them Salesmen in those days), Customer, Area… you name it, we reported it.

There was one particular program that contained a bug. This program had been run in every country, every week, for four years (i.e., over 3,500 times) before being noticed. Cause? Exiting via a CHAIN without a CLOSE resulted in the last block of a virtual array not being written to disk.

(Please appreciate the reports were in 1,000s of US$, rounded, so most managers never tried to reconcile the individual numbers.)

DEC also paid for the development of the “99” reports, which was the original report generator written by the same contractor.

I remember it took five days to train new staff to use the 99 Reports system. The first part was to get them to understand that you can have boxes within boxes (drill down) and boxes by the side of boxes (while/and/not and/or). The hard bit was getting them to understand that the 3rd box down could have boxes side by side. I have to admit that I lost one or two of them from the course. Dennis Mann and I spent a whole day just on that part.

I also clearly remember a strange problem when we were based in Fountain House, Reading, before moving to DECpark. The OAS system occasionally went down in the evening for no apparent reason.

Turned out that it was the cleaner’s vacuum cleaner touching the machines feet and causing a static discharge!

No cleaners were hurt during this misadventure… 

# Eoin Hanley, Systems Optimisation

I started programming in 1967 as part of the Mathematics side of my Physics degree at Manchester and fell in love with it instantly.

My next 5+ years were spent mainly working with ICL 1900/2900 machines in both Cobol and Plan (the ICL Symbolic Assembler) where desk checking and core dump analysis were an essential part of the development process which revolved around the one overnight compilation that was generally possible.

In the summer of 1973, the consultancy for which I worked in Reading, won a contract to supply a warehouse control system for Green Shield Stamps which had just started to morph into Argos. DEC had just opened their first UK office over a sandwich shop (Sweeney Todd's) in St Mary's Butts in Reading. The first UK manager (Geoff Shingles?) was renowned for delivering kit strapped to his motorbike.

Initially, there was a team of four of us. Our first manuals did not have Section 10 (if memory serves me correctly) which covered the use of GET and PUT which are, of course, two rather essential instructions. The reason I mention this is that nobody in DEC's Education Department knew anything about RSTS/E or Basic-Plus. So up pops John Gow (a master of all trades… Sales, DEC hardware, DEC Software who, quite soon afterwards, went on to found Systime) and he did have the appropriate section in his manual.

The design/programming team could only get hands on a machine for more than a few minutes by using DEC's education department overnight in Fountain House, Reading. I would be leaving there at 08:30 in the morning, having been there since 18:00 the night before just in time to see my pregnant wife, who worked for a government department, coming into the same building.

I was responsible for the file design which was somewhat complicated by the fact that we only had two * RK05 drives which had the 'phenomenal' capacity of 2.4MB each and a maximum of 10 disk accesses per second. The warehouse had about 150,000 locations covering 10,000 products and at least 20,000 consignments – and all of this had to be held in less disk space than a quarter of a modern .jpg photograph.

Development was relatively slow with four of us on a single machine but the sheer, unadulterated joy of being able to use a STOP command, examine variables, modify code and then continue the execution of a program still stays fresh in my mind. Relatively soon afterwards, a PDP-11/40, with 28K words or core memory appeared so that we could complete the development. It became apparent though that, with 5 LA30 terminals, the constant swapping of programs was swamping the machine.

The machine was installed, on time, at the Daventry warehouse with the Memory Management module and an extra 20K of Memory on special order from the US. This arrived late one Friday and I had to work through the weekend with Alex Taylor from DEC to redo the sysgen and get the machine up

and running.  We did this by the skin of our teeth.  It is my understanding that most of the software was still running 17 years later!

Here is a snap of our Reading office taken in the middle of the 1980s: please note the piles of paper everywhere and the VT100 graveyard on top of the cupboard on the right.

Thereafter, I was involved with RSTS/E and later the Unix Trans_Basic products for about 27 years.  My final goodbye came around 2002 when a company, for whom I had written a bespoke accounting system got in touch to say that they were being forced by their bank to 'upgrade' to Sage.  Could I please do the data transfer.  The software had been running for 24 years and, apart from me, no one else could understand the incredibly clever four-byte integer functions that had been designed by Eric Bell.

RSTS/E, a wonderful operating system combined with Basic-Plus, an amazing rapid development tool and language, kept me happily in work for many successful years.

*Editor's note: Eoin now has a Husqvarna robotic lawn mower with dramatically more power/memory that the average PDP-11…*

## Errol Ethier, Educational Software Services

### How I came to love and share the love for RSTS/E

It was in graduate school, while pursuing a master's degree in computer science, that I came across this computer company called Digital Equipment Corporation (DEC) whose headquarters were only about 45 minutes away from me.  The graduate school computer I had been using had been a large RCA Spectra 70 computer (an IBM System 360 knock-off), who's only input was

punch cards. Then the computer division of RCA suddenly announced that they were immediately going out of business (that's an interesting story in of itself).

My graduate school had to make some quick decisions. Within a very few weeks we had a large PDP-10 (yes, 10) from this relatively new company, DEC. The PDP-10 range was later re-branded as a DEC System-10 and was promoted as a "Time-Sharing Mainframe".

The beauty of this new PDP-10 computer was that you accessed it via something called terminals – no, not visual display monitors – their general use was still years off into the future -- but converted teletype machines. You typed something into the teletype and miraculously the computer instantly responded back to you, most often with an error message. No more waiting for hours for punch cards to be processed to see if a program you wrote in COBOL, FORTRAN or APL would work. You knew within seconds of typing in a program whether you had correctly written all the computer instructions. The operating system was TOPS-10, a unique operating system which featured something called "timesharing", a relatively new concept. As I soon learned, timesharing allowed multiple people to interact simultaneously with the computer. Where previously my grad school had rows of punch card machines, we now had rows of teletype machines.

(Hey, isn't this supposed to be about RSTS/E and PDP-11's? Hold on… we're getting there.)

What I haven't told you was that I was fortunate to be among the few scores of teachers in the U.S. that had introduced the concept of computers and computer science into a high school curriculum, in my case, a very large high school of over 2,200 students. And my computer science students were punching cards of simple FORTRAN programs for that RCA Spectra computer and when that went away, we had introduced five teletype machines working off the PDP-10 computer from my grad school, working over ever-so-slow dedicated phone lines.

One of the things that was new about this PDP-10 was the use of a new language called BASIC that was very well suited for this interactive environment. Our computer science program, which had started out with about 25 h -selected students (all top math students), grew within the first year to more than 100 students. This created a problem – we needed more and quicker access to computing than the PDP-10 over phone lines provided us. Our success with the rapidly growing computer science program was now a (wonderful) major problem. But the cost of using that remote PDP-10 was very high – we couldn't sustain the increased cost of even more terminals.

What to do to solve this problem? I contacted DEC and asked them about what they might be able to provide us for an on-site computer. Their immediate

answer was a PDP-8, with the TSS-8 timesharing operating system. I researched the PDP-8 and was concerned that it wouldn't be up to the task for a growing computer curriculum. Then a DEC salesperson told me about this new super-duper computer, the PDP-11 and its RSTS operating system, with the Basic-Plus programming language. Living very close to the "Mill" in Maynard, MA (DEC headquarters), I went to see and use this PDP-11. It was instant love! This was a wonderful computing environment, even in its very early forms.

Well it took us several months to convince a school board that we really had to get a PDP-11 computer and for them to agree to pay for it. In the end, about 1973 or so, we got a PDP-11/40 running RSTS/E Version 4B (as I recall). Over time we increased the size and peripherals on that computer including a couple of very over-sized CRT monitors (everyone wanted to use those).

But the kicker to this story is the overwhelming success for many of our students who were fascinated by the RSTS/E operating system and Basic-Plus. In fact, we had to buy several RSTS/E OS manuals since several students were devouring the concept of an operating system and its control of the hardware. Several of our students began writing very-useful system-level programs and some students even learned how to program in MACRO-11. All of this made possible by the unique RSTS/E OS. Many of the ideas for improving the RSTS/E operating system, which I shared over time with the RSTS/E developers, came from our students. Not surprisingly, several of these students went on major computer science colleges and universities in the U.S. To this day, I still have prior students who contact me and relate stories of their experiencing the RSTS/E operating system. Long live RSTS/E (at least in our memories).

## RSTS/E 50th year anniversary… Worth celebrating

It's hard to believe that we're celebrating the 50th anniversary of the RSTS/E operating system. But it's a milestone worth noting. Of course, at this point, it's hard to remember all of the excitement and wonder of getting your hands on – for the very first time – your own PDP-11 running RSTS/E. But I do remember it was an exciting event (much more exciting than today getting the latest generation of new laptop, tablet or phone).

It's also hard to compare the features of RSTS/E that today look somewhat primitive when stacked against the operating systems available some 50 years later. But RSTS/E was definitely a milestone operating system. It was accessible and user-friendly, performing many marvellous tasks but not so intimidating that it prevented you from looking under its hood, so to speak, to understand and start using all of its features (some of them documented and some undocumented).

It was constantly an adventure to learn about new features coming down with each new release of RSTS/E and particularly exciting when, under a non-disclosure agreement, you had the opportunity to meet with the RSTS/E development team who would confirm whether this or that user-requested feature would be available in an upcoming version of RSTS/E. At DECUS RSTS/E SIG meetings (remember those?), one of the best attended sessions would always be the session where the development team would give you a preview of the new features about to be released – standing room only in the session room. There was always an air of excitement (sort of like what Apple now does in its yearly product reveals – "But wait there's more!").

Something I learned about an operating system. While there could be many exciting and useful features embedded in the operating system itself, very often the ultimate excitement centered around the system utilities which matured over time with each new release. Two in particular, I remember, were PIP and TECO – do you remember what the acronym names stand for?

And then I remember the value added to the RSTS/E operating system by technical gurus such as Mike Mayfield who gave us internal insights that vastly increased our understanding, wonder and use of the operating system. Mike helped all of us realize that RSTS/E was magnificently designed.

And, also, there were many technical giants over time that included among others Evans Griffiths & Hart (EG&H) and Peter Dick (wonder what ever happened to him?) of Silver Programs, both of whom developed many wonderful utilities that vastly increased the power, usefulness and life of RSTS/E.

I'm sure I have forgotten to mention others in the RSTS/E user community who made significant contributions to RSTS/E. There were many such individuals and it was exciting to get together with them one or two times a year at a DECUS event. Some of the best learning about RSTS/E occurred in informal meetings around a dinner table or in a session break where you learned about a special feature of RSTS/E or one of its system utilities that you had overlooked. Pure delight!

Yes, RSTS/E's 50th is worth celebrating. May RSTS/E live long in our memories and hearts. For many of us, it was a coming-of-age event.

## It's all in the System Utilities… Live or Die by Them

As I noted in another recollection of RSTS/E, as great as the RSTS/E operating system was, the utilities available in an operating system help make or break the usefulness and value of an OS. At the beginning of its life, RSTS/E had a limited catalog of utilities. In fact, even many of its initial (and future) utilities came from other PDP models (PDP-1, 6, 10, 8, etc.) and/or other PDP-11

operating systems (remember RT-11 and RSX-11?). Fortunately, the RSTS/E development team had a great work ethic over RSTS/E's life: they were never afraid to beg, borrow or steal a system utility from another OS or DEC computer model and incorporate it into RSTS/E. And, as users, we were all the better for that work ethic!

Two very useful utilities that I always remember were PIP and TECO. PIP allowed you to look in detail at and use the file system on storage devices, easily transfer files from one device or directory to another, rename or delete files and directories and other similar activities. An absolutely necessary utility – and one whose features improved over time. Everyone had to use PIP to accomplish daily computer tasks. RSTS/E computer life could not exist without PIP. The commands of the PIP utility were so good in concept and simplicity that PIP and its commands were eventually migrated (rewritten) for other non-DEC early microcomputers (of course, this was all before the modern concepts of drag-and-drop, or copy-and-paste).

Now on the opposite extreme was TECO. This utility was originally created at MIT for use on an early model PDP computer. You had to be a very special (some would say, demented) person to want to learn and use TECO. Among many other features, it was a sophisticated search and replace utility (even more sophisticated than similar utilities available today). For better or worse, I was one of those people who mastered and used TECO magnificently – it saved my own and customers data on several occasions. When I was a co-owner of a company that supported many business RSTS/E systems over a period of a couple of decades, other technical people in our company would ask me to help them resolve a situation where they or a customer had inadvertently mangled key data. TECO to the rescue. TECO was not only a character-oriented editor but also a unique and frequently obtuse programming language – but it was a utility that could perform seeming miracles on data when needed. In fact, TECO was so good (although at times very complex to use) that other computers (not PDP-11s) had user system commands that were translated to TECO commands behind the scenes. TECO also had some sense of humor built in. To create a new text file, you used the TECO command Make [filename]. If you typed, "Make Love", TECO would create the file Love and then would respond by typing "Not War?"

Oh, the good ole days of these system utilities!

Finally… The answers to another article: **PIP** = **P**eripheral **I**nterchange **P**rogram. **TECO** = **T**ext **E**ditor and **Co**rrector.

## Hamish Donaldson, Director Hill Samuel

I first came across Time Sharing Systems in 1966. Dartmouth College had developed their Time-Sharing System and the BASIC programming language on GE 200 Series computers. I was working for Bull-GE at the time and demonstrated possibly the first link in the UK at Herriot Watt University in Edinburgh. It was via a trans-Atlantic telephone line connected to a teletype by modems.

Some years later, in 1973, I joined Hill Samuel and discovered they had ordered a Data General mini-computer as the on-line front end of the new Investments system. A great idea but the machine arrived with very little software so I sent it back and replaced it with a DEC PDP-11 which had "a good operating system" called RSTS (Resource Sharing Time Sharing) programmed in Basic-Plus. This was nearly 20 years before the IBM PC brought computers to everybody's desk and over the next 15 years we introduced PDP-11s to nearly all Hill Samuel divisions in UK, Europe, Australia, Hong Kong, Jersey, USA & South Africa.

I should also point out we had an excellent technical leader in the form of John Sharp who, apart from creating our standard function library for screen handling etc also rewrote the DEC IAM (Index Access Method) to improve performance. John also wrote BASMAC, a Basic-Plus source code to Macro-11 convertor that made our systems go even faster.

Our systems were so advanced that we ended up selling the Hill Samuel Treasury Systems to other banks.

*Editor's note: You can still obtain copies of Hamish's excellent books such as "Guide to the Successful Management of Computer Projects" from places like Amazon.*

## Jeanne Davis, RSTS Product Manager

I was hired at Digital in October, 1980. It was one of the best days in my life! I was hired by Tom Harris, Manager of the Languages and Tools Group, and worked as his secretary for three years. I had another stint in CSSE (Customer Services System Engineering) for a year or so and then, thankfully moved on to eventually land in RSTS land!

I loved the RSTS Group and DECUS trips the best! I had an opportunity to travel to other countries and meet many wonderful people who loved our products.

I got to know and learn from that young group of RSTS engineers. They loved calling me Mom (I think Vernon coined that one), when we were out as a group because I spent my time trying to corral them and keep them out of trouble! Not to imbibe too much as they had presentations or booth duty the next day! They were a handful but we sure did have fun!

The RSTS 20th birthday party was a major highlight for us! Paul and I decided to wear tuxes to celebrate the occasion! We really got into the party planning and had so much fun and delight in celebrating this major milestone.

We all knew that ULTRIX/UNIX was coming along to replace RSTS and RSX, etc. So I became the ULTRIX product manager and began the process of retiring the PDP Languages one at a time....then ULTRIX and its products.

I'm proud and happy with what I learned and accomplished there over those 27 years! I never expected to have a career outside the home but sometimes life surprises us and for that I'm grateful as I would not have wanted to miss out on this wonderful adventure!!!

I retired from Digital/Compaq/HP in 2007 at the age of 60! It was my best decision as the UNIX group was all but disappearing as we transferred everything to India. The writing was on the wall! I raised my hand to retire and take the last good package! I have never regretted that decision and was thrilled that I had more time now to spend with my 10 grandchildren!!! They range in age from 24 to 13, 5 girls, 5 boys! No great-grandchildren yet! We have been very active grandparents and have enjoyed every minute of it!

I have to say that when I left I never looked back. I was very lucky to have met and worked with so many great people! Paul Laba and Kathy Waldron are the two people I have stayed connected with to this day and I feel very fortunate to have them in my life!

*Editor's note: Jeanne is now Mrs Donnelly*

## Jim Condict (Grace), RSTS Development Team

I worked for the RSTS/E development team from 1978 to 1982 under Simon Szeto and Joe Mulvey.

I worked first in the CUSP group and among other things wrote from scratch ATPK and FIT and I eventually became the technical leader of that group.

Then I joined the monitor group and among other things was the technical lead for implementing the fix to the infamous "small buffer" problem and I also became the technical leader of that group.

I represented the development team at a DECUS at the University of Warwick.

It's nice to know that RSTS is still remembered by others.

## John Dawson, HPL Systems, VJV Travel

Please pass on my congratulations to RSTS/E and my thanks to he/she/they for providing paid employment and some fun to so many, including me, and

brilliantly conceived and executed systems to vast, often unappreciative, hordes more at very reasonable prices.

Memories have grown hazy by now as life has moved on. I do recall that much work had to be done overnight and that it mostly involved waiting for something to finish, or just to start. Odd incidents do come to mind...

> Carrying our own mighty 11/34 up a narrow staircase and taking the skin off a couple of fingers.
>
> Long evenings spent writing or correcting code on printouts for typing in the next morning on a VT100 or SYSCOPE with EDT, at first in line mode.
>
> The overnight runs at a Weymouth site which terminated at 2 am when a service lift on the quayside caused a reboot.
>
> Several days attempting to get a printer to respond to a Nixdorf machine. Eventually, as a last shot, soldering together pins 6, 8 and 20. Obvious really, and Eureka! (Or almost).
>
> The engineer who, when asked whether a spare pile of RL02 disks could be put to any use, suggested building a patio.
>
> Discovering background jobs and Send/Receive. Users in ecstasy, unless they wanted to revisit their input within the next ten minutes.

Of course, no blame can be attached to RSTS for the divorces, alcohol and hardware abuse, bad posture and general social weirdness along the way. There was nothing in the code or manuals that required such behaviour.

I now remember RSTS, Basic-Plus and the PDP-11 as being like my Morris Minor - a joy in fine weather and on level ground. Happy days, in retrospect.

Finally a quick history of the tour industry. The version of HPL's RSTS-based tour operating system at VJV (Voyages Jules Verne) moved through the gears of BAS24K, Trans_Basic and other wonders, and progressively from an 11/70 with 80Mb of storage to twinkling Unix boxes with endless Gigabytes. This system was one of the first in the industry to connect to the Web for online booking and finally retired, still developing and as reliable as ever, after nearly 40 years of service. I followed shortly afterwards.

I'll be raising a glass of soda water and muttering a heartfelt `'Hello 1,2'` in celebration after my nap this afternoon.

*Editor's note: John's career included being a "Poacher turned Game-keeper" when he left the software supplier to become general manager of his client, VJV Travel. Small world: the owner of VJV was Philip Morrell and his wife's sister is married to Terry Williams, my ex boss.*

## John Robinson, British Steel, Quantic Computing

It is the early days of 1974, in Middlesbrough, northeast England. A young courting (as it was then known) couple are enjoying a candle-lit evening. He is typing on a Teletype ASR33, working on a computer program. She is playing the game Bulls and Cows, using the program MOO, at another terminal CONNECTED TO THE SAME COMPUTER. They are in a nondescript industrial unit and are candle-lit because the United Kingdom government, in the wake of oil price rises and with a coal miners' strike threatened, has severely rationed the use of electricity. Computers could be powered but not the lights in the room.

On the day the "Three Day Week" was announced on 13th December 1973, I did a SYSGEN for Version 5A: GDS stands for General Steels Division, part of the British Steel Corporation.

```
$CH SYSGEN.BAT^U
.RUN PIP
ILL CMD!
$RUN PIP
ILL CMD!
$CH SYSGEN.BAT
ILL CMD!
$CO
$RUN PIP

A002 046600
$CO
$RUN PIP
$RUN CILUS
$RUN PIP
SYSGEN:WHEN SYSLOD IDENTIFIES ITSELF,
SYSGEN:MOUNT A NEW DISK ON UNIT DK0:, WRITE ENABLED
SYSGEN:RESPOND AS FOLLOWS TO THE PROMPTING:
SYSGEN:DIALOGUE? <RETURN>
SYSGEN:#DK:/NS:256:49/F0/H0/B0<DT:RSTS.LCL
$RUN CILUS

SYSLOD V06-06
DIALOGUE? NO
#DK:/NS:256:49/H0/B0/BL:226<DT:RSTS.LCL
SYSLD COMPLETE


RSTS V05-21 GSD RESEARCH

OPTION: RE
DD-MMM-YY? 13-DEC-73
HH:MM? 13:04

OLD ? Y


OPTION: DE

NO DEFAULTS ARE CURRENTLY SET.

YOU CURRENTLY HAVE: JOB MAX = 16, SWAP MAX = 8K

JOB MAX OR SWAP MAX CHANGES? Y
SPECIFY A NEW JOB MAX? NO
YOUR NEW SWAP MAX IS? 14


CURRENT MEMORY ALLOCATION TABLE:

 ADDR   +00000  +04000  +10000  +14000  +20000  +24000  +30000  +34000
000000    MON     MON     MON     MON     MON     MON     MON     MON
040000    MON     MON     MON     MON   BASIC   BASIC   BASIC   BASIC
100000  BASIC   BASIC   BASIC   BASIC   BASIC   BASIC   BASIC   BASIC
140000  BASIC   BASIC     U       U       U       U       U       U
200000    U       U       U       U       U       U       U       U
240000    NXM     NXM     NXM     NXM     NXM     NXM     NXM     NXM
300000    NXM     NXM     NXM     NXM     NXM     NXM     NXM     NXM
```

CONTINUE

And here is a typical example of our professionalism in those days and please note I was allowed a massive 100 disk blocks of storage!

```
HELLO

RSTS V05-21 GSD RESEARCH  JOB 7  KB9  18-OCT-74  15:42
#16,10
PASSWORD:
**NOTICE   15:06   18-OCT-74
        JOHN ROBINSON WILL BE AWAY FOR A WEEK.
HIS PLACE ON THE STUPID QUESTION ROTA WILL BE TAKEN BY:
        STEVE GRAY      MON,TUES
        D. A. PEEL      WED,THURS
        F. SHIPTON      FRI



READY

BYE
Y
CONFIRM: SAVED ALL DISK FILES; 98 BLOCKS IN USE, 2 FREE
JOB 7 USER 16,10 LOGGED OFF KB9 AT 18-OCT-74 15:43
SYSTEM RSTS V05-21 GSD RESEARCH
RUN TIME WAS .4 SECONDS
ELAPSED TIME WAS 1 MINUTE, 1 SECOND
GOOD AFTERNOON
```

The computer is a PDP-11/45, made by the Digital Equipment Corporation. It had been delivered a few months before to the newly established Research Laboratories of the British Steel Corporation. The operating system, which allows multiple simultaneous users, is RSTS/E. The PDP-11 and RSTS/E is to help shape the next 15 years of the couple's lives. [Last year they celebrated their 45th wedding anniversary.]

The time-sharing system and its interactive Basic-Plus programming language, allowed dozens of researchers to develop their own programmes for mathematical modelling, data analysis, scheduling etc. Maintaining and extending the system taught me a lot about operating systems, interpreters and hardware. We added a pen-plotter and soldered diodes onto a circuit board to adapt the boot-rom code for a new disk drive. And there was a lot of fun: VT50s (early CRT text screens) could play a version of Space Invaders, a colleague wrote a 'time' program with messages like 'lunch was half-an-hour ago' and 'just coming up to tea-time'. There was another PDP-11 powering the GT40 graphics terminal, with a lunar Lander simulation, which had a 'McDonalds' Easter egg – but none of us knew what a 'Big Mac and a Cheeseburger to go' meant.

In time the 11/45 was moved to a new purpose-built research centre. The office manager found the cheapest removal firm he could – a guy turned up with a Land Rover and a horsebox. We said NO! The 11/45 gave way to an 11/70, with 4 megabytes of RAM and massive RP04 disk drives, 88 megabytes, the size of washing machines – the laptop I am typing this on has 5000 times that amount of storage.

After some years I and my boss (David Peel) left to form a small consultancy, Quantic Computing. This was a partnership and David (also a mathematician) suggested the name from the term for a 'rational integral homogeneous function of two or more variables'. By this time, RSTS/E was used in a lot of commercial environments. I got to visit places other than decaying steel-towns: I was impressed by the art deco, Grade II listed Gillette factory in Brentford but not so much by the scruffy unit in Glasgow where top-end hi-fi systems were designed. The firms included one car spare parts shop run by a retired Formula One driver and another by a famous rally driver. I also went to couple of DECUS conferences in the south of France; you know a company is doing well when the default drink at the reception is Lanson Black label champagne.

At the Research Laboratories we had delved into the Basic-Plus system. This used PPCODE, a byte-oriented abstract code executed by a virtual machine [a technique used in the later languages Java and Python]. For scientific use we added extra mathematical and string functions and a nested error-handling scheme.

I met Peter Dick at the DECUS UK conference in Exeter in April, 1984 and discussed an idea for extending the limit on the size of user programs. June that year was spent in an intense effort, working with Steve Gray. In two days we showed splitting the Basic-Plus support 'Run Time System' could be done; in two weeks we could run programs 50% larger than the original 16Kword limit. Another two weeks of testing, checking build options and refining: and Peter had a product to sell - BAS24K.

*Editor's note: this "50% larger" is slightly misleading. Programs hitting the 16KW maximum program size already had all their files open, initial system overheads and standard functions and so the increase from 16KW to 24KW allowed for more than 100% increase in specific code within a program.*

*Author's note on Editor's note: I love that nearly 40 years on I stuck to the technically precise evaluation but the Editor highlighted the enhanced worth to the user. Ever the fine salesman!*

What made RSTS/E so enjoyable and rewarding to work with? It was developed by a corporation led by engineers with primarily technological rather than financial goals. The hardware was becoming cheaper and more reliable (a user reported a 'bug' with the SYSTAT uptime overflowing at 999 hours), allowing computer use to spread to a bigger market of small and medium sized businesses. But hardware and software had not yet become standardized and commodified, so there was the need and scope for small teams to develop new uses and applications.

I was lucky to be working at this point in computer development and to be involved with such a great system: thank you RSTS/E.

## Jon Power, Sector 7

My strongest memory from those early days was the DecUser Magazine Challenge issued by Peter Dick in August 1987.

Perhaps I should point out that, by then, DEC had published their 900 page RSTS/E and VMS Compatibility Guide … or should that be Incompatibility Guide.

An early entry in a USA software directory states:

*Sector 7 is a full service software house. We can supply tools or tools and services. We can take your application in-house port it and deliver a working system back to you.*

*Since 1985 Sector 7 has written hundreds of API's and conversion tools targeted at significantly reducing the time it takes to port an application. By using our tools, the time, cost and risk associated with a migration or port can be reduced by 90%.*

*We can supply experienced people who know how to plan and execute a migration or we can just be there when you need us. Just 1-800-VMS-UNIX-NT for the first step in rightsizing YOUR application.*

So… the challenge was to migrate one of his (complex) RSTS/E Basic-Plus programs to run under a different operating system. Peter was convinced that nobody would achieve this and said he would “eat his hat” if anyone could do what he thought was impossible.

Wrong!

23 software migration packages applied a copy of the code and about three or four of us succeeded: apart from our own BTRAN, there was C-GEN and Trans_Basic. I can’t remember the last one (KBasic perhaps??).

Sector 7 went on to specialise in migration and eventually merged with I.B.M. with whom we still work.

Finally… one more memory… the pictures of Jean Davis, DEC RSTS/E Product Manager, in Las Vegas with the Chippendales cast… along with the, as usual, worse for wear RSTS Users…

# Ken Surplice, DEC Sales and Marketing Wizard

Life on the inside
*Ken Surplice, one-time DEC Software Specialist and DECrep*

They were good times. They were simple times. They were passionate times. They were RSTS/E times.

My first job, lucky me, was the role of “Software Specialist” in the Reading branch office of Digital Equipment Corporation. This was back in the 70s, not so very long ago. I remember two particular moments in my early days.

The first was the boss asking me to call a customer about their problem. “But I don’t know anything!” I exclaimed. “You’ll be fine" was the reply. He was right.

The second special moment was when management informed me that I was to be the DECrep for the UK RSTS/E group. “Who are they?” I exclaimed. “You’ll find out” was the reply. He was right.

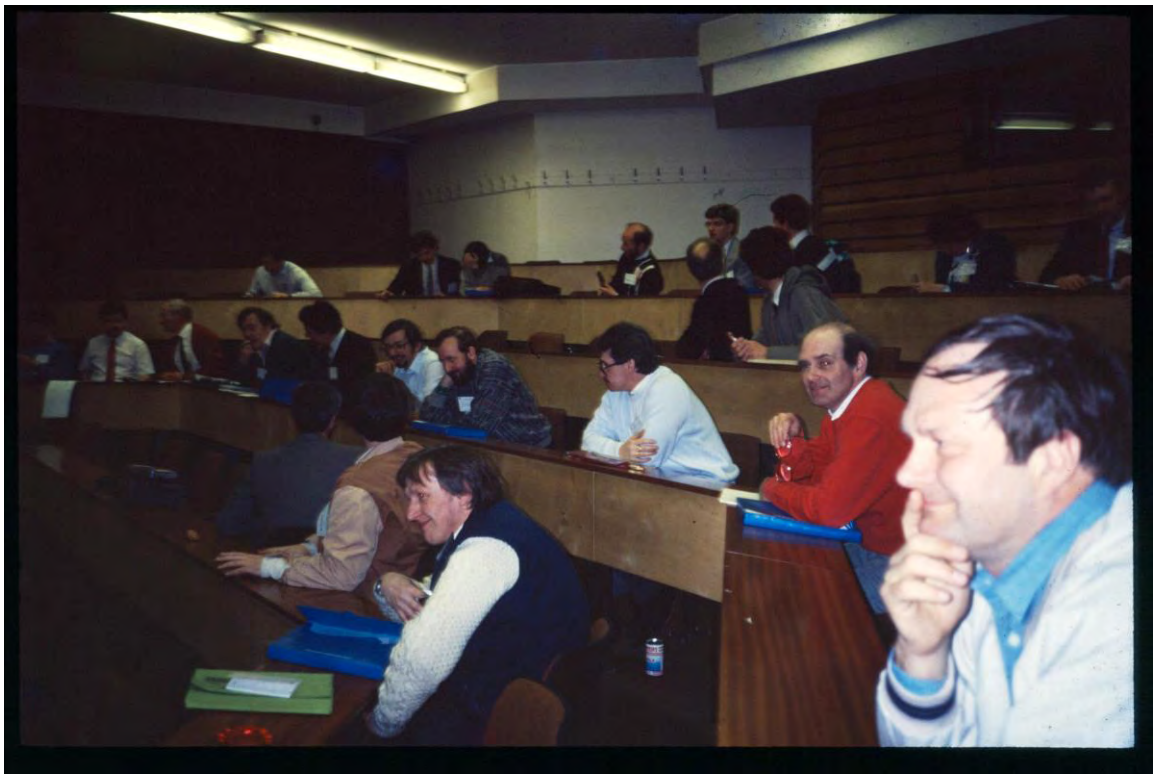
So started a lifetime of engagement with computer user groups. Was being a DECrep akin to be thrown to the lions? Don’t tell anyone but no, not really. It was an honour to represent our users and try to get the rest of Digital to support them well. “Users talk, RSTS listens” as the t-shirt said. This was possible because in these times we knew who “RSTS” was. We had names. We DECcies knew all our branch colleagues from Field Service through to logistics. We even knew the names and faces of our very own personnel reps, who were in the same building. Of course, we didn’t know marketing. They lived in HQ in Reading, which turned out to be a land far far away, as we’d moved to Basingstoke and we didn’t have email or other means to engage. We had freedom. We were self-sufficient. We knew our customers and they knew us. Perfect.

Do remember the early days of RSTS/E. There was no paid support. You just expected it to work and picked up the phone if it didn’t. There was no automated patching. Usually you read the Software Dispatch and typed in the patches. If you ever thought airhostesses and stewards were simply there to serve you refreshment, I can forgive you for thinking that the role of a Software Specialist was to answer the phone and hear you thunder down the line “where is my Software Dispatch?”

Who were these users? In the early days, users came in different breeds. There were RT-11 users. There were RSX-11M users. There were IAS users. There were MUMPS users. There were runtime systems RSX-11S and MicroPower Pascal. And there was RSTS/E. Incredibly, all these different

software environments ran on just the PDP-11. The Hitchhikers Guide to the Galaxy describes Earth in the same way I could describe RSTS/E users: “mostly harmless”. At the same time I could recognise that they were passionate about their software, they were ~~rich~~ very rich in character and they were our customers, ensuring that I got paid and could buy them their next beer, or even crates of beer as once was the case at a DECUS user group conference. Well, you have to try, right?

This is what my bosses saw: RSTS SIG at DECUS UK Warwick 1988



This is what I saw: Still RSTS SIG at DECUS Warwick 1988


Then along came VAX. Virtual Address eXtension, running Virtual Memory System. This was a disaster, at least from a naming point of view, as I had proposed ERA machines running EROS. That would be Extended RISC Architecture running Extended RISC Operating system. As VAX was calling, naturally Digital wanted customers to keep buying PDP-11 computers, later to morph into "servers" and RSTS/E licenses, while VAX sales ramped up as applications were ported. When the fashion is to buy off-the-shelf applications, getting your application provider to support your new architecture can be ~~quite a struggle~~ expensive. We should have stuck with Basic-Plus, written our own code and avoided being at the mercy of others. Well, that didn't work, did it?

Fortunately, it turned out that customers wanted to keep buying PDP-11 computers anyway and continued to grow the RSTS/E population, for quite a long time. Let me confess. At home, I still have reams of paper with a letterhead proclaiming PDP-11 commitment, long life and something else. I can't remember where I keep it so I can't check but the long life bit was a commitment we kept. While VAX moved to Alpha, sigh and then to Itanium, a good friend of mine still makes a living configuring and supporting PDP-11 machines. It's all good stuff but for many the cost of change, or perhaps the fear of it, exceeds the cost of extend support for the good stuff they have built up over the years.

All this is a far cry from the situation today where operating system developers and user groups alike mean groups of thousands of people. The direct connection that we had has gone. While I miss the personal touch of RSTS/E

and its users I feel privileged to have ~~beer~~ been there and done that. The older I get the better it was.

Happy 50th birthday RSTS/E.

Ken Surplice
Ex-DEC rep and happily retired from the computer industry

## Kevin Galloway, Sector 7 UK

Gosh, stories, I'd need to trawl the archives of my brain but I'm sure I could come up with something!

What was it at the sig meetings, pirates and victims or something like that?

Or how many times did we do a migration and the client *said "its not working right, our overnight run took eight minutes instead of eight hours"* - think that might have been UBS.

## Mark H Bramhall, RSTS Development Team

It is hard to believe that it has been 50 years.

I graduated from Carleton College in June 1970 and joined DEC in July. I was in education institution marketing for maybe two years. Sold RSTS. It didn't work. We always had to add more memory. So, I switched to dev to make it work.

I carried one of the earliest deliveries of RSTS in 1970 to Carleton College, Northfield, MN, USA… the very same college from where I had graduated!

Installations in those days were all custom generated for each customer in Maynard. And cross-compiled on TOPS-10, the DEC in-house timesharing "main frame", then punched to paper tape.

I carried it as a paper tape in my briefcase. There were no utilities at that point. I wrote the first SYSTAT on site just to see what was going on...

I would also like to mention some aspects of the internals of RSTS: here are three memories about Garbage Collection, the CAT command and the "erase on delete" file attribute.

### Garbage Collection

When a new string is needed in a Basic-Plus program and there is not enough free space, the garbage collector (GC) is called. The Editor's opening paragraphs contains the sentence:

> *RSTS/E did lose some oddball things, such as the fact that RSTS-11 did Basic-Plus string garbage collection through a file processor overlay*.

This apparently harmless summary caused a flurry of emails between Paul Koning, Steve Gray and myself. Paul found my original code/comments, on the web, from 1972: I am the "MHB" noted in the code…

```
COPYRIGHT 1971,1972 BY DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
VERSION:      04A           EDIT:   79
DATE:         01-SEP-72     BY:     MHB/JDM

WRITES A FILE OF ACTIVE STRINGS AND READS IT BACK THUS ELIMINATING
"HOLES" IN STRING SPACE.

FIRST PASS MOVES SIGN OF FIRST WORD OF STRING TO SIGN OF LENGTH WORD
IN HEADER AND CLEARS SIGN IN STRING. THIS MAKES THE STRING SIGN
AVAILABLE FOR USE AS A FLAG.

THE SECOND PASS SETS THIS FLAG TO INDICATE THAT A STRING IS POINTED TO
BY A HEADER, AND USES IT TO SET THE "MULTIPLE-REFERENCE" FLAG IN ANY
HEADERS WHICH POINT TO STRINGS WHICH HAVE BEEN POINTED TO BY EARLIER
HEADERS.

ON THE THIRD PASS THE STRING SIGN BITS ARE RESTORED AND THE STRINGS
ARE WRITTEN IN A DISK FILE.  THEN THE NEW ADDRESS OF A STRING IS
STORED WHERE IT USED TO LIVE. HEADERS WITH THE "MULT-REF" BIT SET PICK
UP THE NEW STRING LOCATION FROM THE OLD STRING HOME.  THEN THE DISK
FILE IS READ IN OVER THE OLD STRING AREA.

THE GARBAGE COLLECTOR WORKS ON THE PTA AS WELL AS THE PDA.

SINCE THE GARBAG COLLECTOR IS LONGER THAN 256 WORDS, IT READS THE
SECOND HALF OF ITS CODE INTO THE FIP DIRECTORY BUFFER BEFORE DOING
ANYTHING ELSE.
```

On the other hand, when RSTS-11 evolved to RSTS/E, garbage collection became a compute-only, memory-only routine that shuffled in-use strings together thus leaving a bigger free space. If the string still could not be allocated and if the "low seg" (the user's r/w area) was not at its maximum, an exec call was made to expand by 1K (words). This exec call might well cause swapping – of either the caller or others. If the caller could not expand in place, swapping reshuffled memory to permit the new larger size. So, I can see someone easily seeing swapping while running some ever-growing program.

At some stage in the early 1970s as "being green" started to become popular and bean counters started to influence engineering, the GC (Garbage Collector) was renamed the Economizer.

## The CAT Command

We need to discuss FIP before looking at how CAT worked.

The file processor (FIP) was blocking and single stream. Any FIP call would make the user not runnable (until the call completed), but swappable, and the call would be queued to FIP. When FIP was waiting for disk I/O, other jobs could run if they were runnable. There were two buffers used by FIP. Both were permanently allocated in low memory. One was the overlay buffer. This was only read into for whatever overlay FIP needed. The other was the directory buffer. This was highly r/w. It was read into, modified, and then written back. All FIP calls were atomic – that is, there was never FIP state outside of FIP.

For the CAT command, the call was "give me the nth file in this directory". The caller (Basic-Plus) kept track of the n. So, when the nth file was asked for, FIP had to read the directory entries for entry 1, 2, 3 …, all of the way to n. If the directory was not compact (and it was almost never compact), this meant many disk reads. Before the disk-caching feature, these were all physical reads. So, a CAT command could really rattle the disk. And, therefore, cause a large FIP queue backup.

### "Erase on delete" file attribute

The longest possible FIP operation was deleting a file that was marked "erase on delete." Such a file could be huge, especially when the "large file" configuration option was enabled and FIP would use that single directory buffer, fill it with zeroes and write it, one block at a time, to the entire file. We calculated it could take up to 4 or 5 hours of such writing for a suitably large file. And, yes, FIP would block the entire time!

Impossible as it may seem, I am still creating software. I am full time employed by a small hardware/software company based in Paris. I would be there now except for getting back to MA to be with my wife during the Covid-19 lockdown.

## Mike Markowski, University of Delaware

Thanks to the wonder of the Apollo program as a child, I loved math and science. Later on, computers were science fiction come to life after many Isaac Asimov books! They fascinated me and in 1977-78 in 9th grade I learned that Computer Science would be offered the next year in high school.

I excitedly signed up and learned Basic-Plus on a PDP-11/70 running RSTS/E.

After school, I played trek and adventure on a DECwriter, killing many trees, wrote programs and learned RSTS/E.

Once, I broke into a privileged account - oooh. I didn't know what to do and did nothing bad but it was exciting. I was eventually brought on-board as a student programmer for no wages and never thought twice about it. I was happy not to have to *pay* for the privilege.

When University of Delaware Project DELTA lead, Ed Boas, gave me a RSTS/E Basic Programming manual, there was no higher honor.

I was happy to be part and after earning my driver's license drove 1/2 hour to University of Delaware where the DECwriter was replaced by HP2621 crt terminals. So wonderful! To this day, certain songs from the 70s, like 'What a Fool Believes', bring me back to days of being a student programmer comfortable with TECO and RSTS/E!

Now, I work in an RF lab with radios and program in python on linux machines. The job is a perfect fit for my hardware software interests and built on a foundation from 1978 learning RSTS/E!

(256 words of some memories, as per the original email request!)

*Editor's note: here is the link to Project Delta.*
https://udel.edu/~mm/projectDelta/

## Mike Mayfield, NDS (Northwest Digital Software)


*circa 1975*

Mike Mayfield
45712 SE 138th Ct
North Bend WA 98045 USA

What a blast from the past to look back on our days with RSTS and everything it meant to so many people over the years.

So much of my early adult life revolved around that frustrating and fantastic operating system, RSTS/E. My first job, right out of college, was with Digital. I was 21 and RSTS was only 3 years old. I specialized in RSTS for over 20 years, so I guess you could say we grew up together. As did my children:

Looking back, it is absolutely amazing what we were able to do with a PDP-11 and RSTS. The smart watch on my wrist has almost 200 times the CPU power and 6000 times the memory as the best PDP-11. But, with that great old computer we were able to support 63 simultaneous users, running everything from payroll to word processing and maybe even a game or two thrown in.

Some of my favorite memories are of hanging out at DECUS with my friends Anton Chernoff, Paul Laba and Jeff Killeen, etc. You just can't beat walking around town with Bozo the clown after Jeff or Paul (I can't remember which one) came out dressed as a clown when someone complained that "any Bozo could create an index for RSTS docs."

During those 20 years with RSTS, my business partner, Mark Feniello and I were able to create some pretty cool products that I'm really proud of but most of them have some little known history that I would like to share:

WORD-11 was the most popular word processing system in the DEC market but it was invented on a backpacking trip to the Sierras. DEC had recently released the WPS-8 word processing system that used a dedicated PDP-8 computer per user. While I was walking along, I figured out how to use RSTS's character buffering feature to allow this to work in a timesharing system. I brought the idea to DEC but they said it would never work, so I left DEC and partnered with DPD to create WORD-11.

SAVER was a popular backup package. It was the only backup product of the time capable of streaming to a tape drive. What most people don't know is that it was created over a weekend as a birthday surprise for Richard Marino, DPD's product manager.

The most difficult bug I have ever faced was in The LINK, a program that let you cluster up to four PDP-11s running RSTS into a computer that outperformed the fastest VAX. We had a bug that caused a crash but only happened on a handful of computers, all in England and only after about a week of uptime. Peter Dick and I worked together for seven months to track it down. I eventually went to England and stayed with Peter at his home, Abbots Dwell, while we worked 20-hour days for almost a week before finally vanquishing it. He still reminds me about my giving up and going to bed, only to reappear 10 minutes later with the fix.

RSTS Performance Monitor was a performance-tuning tool. It helped a lot of people optimize performance on their systems but the thing I like best about it is the name, RPM. What a cool name for a performance tool.

I have written three books over the years but my favorite memory is getting a large order from DEC for my RSTS/E Internals Manual. They created this great

operating system and here they are ordering this book I wrote at my kitchen table in the woods of Washington. It's a strange world.

Finally: the RSTS Ballad https://www.youtube.com/watch?v=AWPOp8vjRQY

*Editor's note: Mike has omitted his most famous program from his already impressive list of software products.*

*In 1971, while a senior in high school, he wrote the original Star Trek game in BASIC on a Sigma 7. This text-based game ran on an ASR-33 Teletype at 10 c.p.s. David Ahl (DEC Educational Services) included it in his book "101 BASIC GAMES", published by Digital in 1973.*

*Steve Wozniak, in his biography "iWoz" published in 2013, credits wanting to be able to play Star Trek as the reason the original Apple I computer ran BASIC instead of (say) FORTRAN. Steve also says the first program he ran to test his version of BASIC in 1976 was Mike's Star Trek.*

```
1   REM ****  HP BASIC PROGRAM LIBRARY  ******************************
2   REM
3   REM       STTR1: STAR TREK
4   REM
5   REM       36243  REV B  --  10/73
6   REM
7   REM ****  CONTRIBUTED PROGRAM  **********************************
100   REM ************************************************************
110   REM ***                                                        ***
120   REM ***     STAR TREK: BY MIKE MAYFIELD, CENTERLINE ENGINEERING   ***
130   REM ***                                                        ***
140   REM ***        TOTAL INTERACTION GAME - ORIG. 20 OCT 1972
150   REM ***                                                        ***
160   REM ************************************************************
170   GOSUB 5460
180   PRINT "                              STAR TREK "
190   PRINT "DO YOU WANT INSTRUCTIONS (THEY'RE LONG!)";
200   INPUT A$
210   IF A$ <> "YES" THEN 230
220   GOSUB 5820
230   REM *****  PROGRAM STARTS HERE *****
```

## Mike Norris, Systime Field Service Engineer

When I started in the Computer Industry (October 1978), it was with a Leeds (UK) based company called Systime. We did well competing as a DEC O.E.M. against DEC's other O.E.Ms in the commercial mini-computer market. At one stage Systime was ranked at the U.K.'s largest computer manufacturer, by volume. ICL was still larger by value.

Systime had a number of tricks which they utilised to give them the edge, some not entirely legal leading to a licensing case later when Digital took them over (but that is another story)!

Practically all the systems we sold in the early years ran RSTS/E, the systems based on the 11/04 were marketed as 1000 and 3000, those using the 11/34 were 5000's.


The first trick they used on the 1000 systems was to take the bottom 2K (words) of the I/O page and map it into user space, giving the user 30K rather than 28K this was achieved by patching the Operating System RSTS Version 4. RSTS Version V could support a number of users depending on the job space allocated but performance suffered particularly on the (then) slow disk transfers, especially when the jobs started to 'Swap out'.

The disk subsystem was quite often using Drico 2.4 Mb drives which employed a rack and pinion type positioner, being visible to the system through a copy (almost identical!) RK05 disk controller. So to speed this up Systime introduced the 3000 machine, again this ran RSTS Version 4, it was also an 11/04 based system to which they added a memory control module and a special backplane this allowed direct access to 128K (256 kb) of memory, it then swapped directly to memory rather than to disk providing a serious improvement in performance.

This was again achieved by patching RSTS. It led to a real selling advantage, especially in the early years when the cost of the machines were so high and small and medium businesses were really pushing to get a 'Computer'. On top of this Systime built a range of commercial software packages, particularly focused on these businesses for accounts and invoicing, so were able to provide a complete package and became a real thorn in Digital's side for a number of years.

Thanks to the versatility and logical design of the RSTS/E Operating system all this came to happen!

And… I am delighted to say I no longer pick up the phone, fearing a customer saying "Help Mike. We've got this message: It says… `Device hung or write-locked` and we've got to get the payroll run today…"

## Moshix, mainframe channel on YouTube

In 1981 a dear friend (who sadly died shortly after) and I took our bicycles and cycled for two hours to go to the computer fair in the New York City. We were really interested in PCs with CP/M. But we went to the DEC booth and a very lovely young woman with no customers keeping her busy asked us to sit down at a terminal and learn about RSTS.

We did and she explained to us the editor (I was amazed at the complexity, coming from WordStar) and some of the commands of RSTS. We liked the young lady so much we kept asking questions. Finally, she said the fair is closing doors and she needs to shutdown the PDP-11.

I remember still clearly her scent sitting very close to me. I was 16 years old and it was the golden era of computing.

I (also) fell in love with the DEC amber terminals. I recently bought myself an amber VT320 and connected it to my emulated VMS system. Mostly because of this memory from the computer fair.

## Nick Brackenbury, DEC Salesman, Darkcrest

HOW GREEN SHIELD STAMPS MORPHED INTO ARGOS…


The story starts in the early 1970s, possibly 1973, when Green Shield Stamps was big business. Argos was just being planned.

I was invited to a meeting with an IT Man and, I think, the son of founder Richard Tompkins. They wished to purchase a minicomputer to support three or four terminals initially.

The application was stock control for a new warehouse in Daventry. It was a straightforward multi-user minicomputer requirement. We got on well with good discussion on their needs.

There was a 'budget hick-up' as always. Compromise was RAM size which worried me. They insisted that the application was very simple.

My competition? Cannot remember, usually HP (too expensive) or Data General (low price means dodgy …ha ha).

From programming viewpoint, the most suitable O/S would have been MUMPS with its natural database. To manage the workload, RSX11-M is most efficient run-time but hardest to program. But well, RSTS was DEC's official product for Multi-User Business systems. Basic-Plus was gaining popularity with programmer community. So I quoted PDP-11/40 and RSTS.

Hoskyn's would have done a MUMPS System. Logica would have implemented RSX11-M system. But somehow some entrepreneurial software guys muscled in pushing RSTS…

I innocently asked them, "You are a Trading Stamp company, why do you want stock control?" If I had been a Reporter I could have made a killing with what they told me! Stamps were given away in petrol stations to attract more clients. The stamps were later redeemed for goods in shops that accept them.

Green Shield learnt all about high street product trends. They could watch High Street goods movement trends. Price, quality, volume, seasonal trends. Everything.

Green Shield realised that they could launch a cut price chain of shops selling only the right products and always having the best price. Now look at the size and strength of Argos!

Close to 50 years old as well.

*Editor's note: I remember Nick telling me about a subsidiary company called Fungus. I confessed that I didn't think much of the name. Nick said it would grow on me…*

## Nigel Williams, Hobart Tasmania, Australia
## www.retroComputingTasmania.com

As a teenager still at school, I had the chance to play on a PDP-11/70 running RSTS/E at a local higher-education college back in the late 1970s. The college was known as TCAE - Tasmanian College of Advanced Education, located at Newnham in northern Tasmania. It had multiple campuses across our island state. There were about a dozen terminals connected from across the state to this PDP-11/70.

After work hours I was allowed access to a couple of terminals from my location at the Mount Nelson campus (now the home of Hobart College), about 220km distance from the 11/70. One of the terminals I could use was an LA36 DECwriter at 300 baud and the other was a CRT terminal running at 2400 baud. Even with the distance but likely helped by access outside business hours, the system was responsive and interactive. I remember feeling at the time that the look of the LA36 was like living in the future -- what I would now call design aesthetic -- it looked so sleek and purposeful, a sports saloon of terminals[1]. In contrast, the CRT display terminal was a clunky black metal box but with a faster screen refresh and so much nicer to work on. These sessions were

highly prized by me, as I had access to a large minicomputer system and the luxury of two different terminals to tinker with. I would only get a short time with the system between leaving school around 4pm, however, limited on both ends by a bus journey eating up precious time and being kicked out by the security guard around 6pm.

The RSTS/E system ran a student records system and was also available to students for learning and teaching. The student records system was developed by a couple of fellows in the college computer section, the late Graham O'Connor and his boss whose name I can't recall. It was a combination of Basic-Plus (BP) and COBOL programs. The BP application was a screen-oriented CRUD-style design (Create Read Update Delete on indexed files) and the COBOL suite did the batch-processing component. Looking back, I am guessing it was RMS-11 under the covers, as that supported indexed files that could be shared between BP and COBOL.

The BP program was massive and really stretched the capacity of the system. I recall Graham constantly struggling to squeeze in more functionality and scrabbling for every word of memory. This data entry and enquiry application had an elaborate screen handling mechanism to paint the screen with data-entry forms and provide what was then a relatively user-friendly experience to the operators. Much use was made of the simple cursor control and line editing features. The page was drawn and the cursor positioned at the entry point; fields were annotated and repainted as the user navigated through the form. This was for the time an innovative approach to user interaction, particularly given the low line speed, limited CRT display control and lack of system support for a forms interface (all of which developed rapidly in the 1980s). Some years later it was replaced with various VAXen and much of the application software was migrated onto the new platform.

My main contribution to this RSTS/E system was a password changing program. Being young and naïve, I neglected to ask users for their current password before offering to change to a new password. Within hours of it being released I was horrified that the students were changing each other's password as a prank and I had to hastily add a “what is your current password” question before allowing the change. Being allowed to do a FIP-privileged system call like

```
A$ = SYS(CHR$(6%)+CHR$(8%)..."NEWPASS"...etc.)
```

felt like being handed the keys to Dad's car.

At some point another terminal was set up and connected to the remote 11/70 but in a different building on the campus. To my delight I was given out-of-hours access to this building. I could while away time tinkering on the 11/70, usually on weekends when the building was empty and with no time restrictions

as to how long I could spend there. This multi-storey building was a bit odd in that many of the rooms (including the terminal room on an upper floor) had no windows.

One weekend, while using the system, the power went off, followed by an enormous bang. Sitting there in the pitch black, with a now-dead terminal, I was wondering what to do. I was a bit concerned that something dire had happened (like a fire) and I should be evacuating the building. The fire alarms had not sounded, however, so I felt somewhat less urgency. The loud bang was disconcerting and I was somewhat torn whether to stay where I was. After several minutes of waiting, I thought I better leave the building and hunt down the security people to see if the power outage was local or widespread … there were of course no mobile phones back then! I exited the terminal room into an equally pitch-black corridor, and turning sharply left heading towards what I thought was the stairwell, promptly slammed into a solid wall. Now I started to panic. There should not be a wall here. I felt back to the door into the terminal room and tried to figure out how I could have turned in the wrong direction. In the darkness, I was sure I had made a mistake. I retried and still came up against the wall. Confusion reigned. I am not sure how many times I felt my way back to the terminal room doorway but more than a couple I am sure.

At some point I recalled the loud bang after the power went off and had a dim memory of a sound of something heavy sliding shut before the bang. My brain started to join the dots and I finally realised that the corridors had self-closing firewalls that rolled shut at loss of power. I felt to the end of the "new" wall and wedged my fingers into the join and after a bit of effort was able to force it open and slip through a gap and out into the stairwell and daylight. The panic had completely swamped my ability to wonder at the time why the corridor was pitch black, as it opened onto open daylight stairwells at both ends. I think it took a few weekends before I ventured back to the terminal room.

The TCAE PDP-11/70 was one of only two in the state used in education. The other was at the Elizabeth Computing Centre (ECC), also running RSTS/E. This system provided a statewide teaching environment, which I used in one of my college computing courses. At some point I was given permission by ECC to borrow the RSTS/E and BP source-code listings for a few nights to look at. I lugged the huge printout binders (30 cm stack as I recall) all the way to the bus stop, stopping frequently on the way to shift and relieve the weight of all that paper. I spent the next several nights pouring over the MACRO-11 assembly language code listings and remember my excitement at discovering the push-pop code mechanism buried within Basic-Plus. I had spent a previous summer laboriously hand-disassembling SYM-1 BASIC for the MCS6502, a straight interpreter. I was fascinated to see a completely different way of building a BASIC interpreter/compiler by comparing BP and SYM-1 BASIC.

Several years ago I went looking for the BP decompiler that reversed the push-pop codes back into source-code:

https://marc.info/?l=classiccmp&m=126639435321604&w=2

```
[80,19] BASIC-PLUS-1 DECOMPILER: two programs, one
data file BASIC-PLUS-1 reverse compiler

I have two of the three pieces kindly sent to me by a
contact on the Deltoids mailing list, I have UNCMP.BAS
UNCMP2.BAS but missing the UNCMP.DAT file.
```

If anyone reading this has the missing UNCMP.DAT file I would be glad to receive a copy as that would complete this RSTS/E artefact.

My encounters with RSTS/E and the PDP-11/70s set me up for a life-long career working with computers. At the time I was using these machines I thought it was impossible to have a career in computing. I was just as likely to get a job as an astronaut. But I nerded my way into a computing career eventually and never looked back -- but I did study accounting in college, just in case!

[1] https://user.xmission.com/~legalize/vintage/images/dec/la36.jpg

## Noel Chiappa, Curator of PDP-11 History

RSTS-11 was the system on which I learnt to program.

Our math teacher at my grammar school in Bermuda, (very roughly comparable to a UK grammar school; it went up through 'O' levels), had done a little bit in class in programming but it was purely a paper exercise. When I went away to school at Phillips Academy in the fall of 1972, their math department had a PDP-11/20 running RSTS-11, which they had gotten a year or two before I got there.

During my first week, I borrowed a Basic-Plus manual from another student (who was there the previous year and was already up to speed on RSTS - I joined half-way through a 4-year US high school program), and learned to program by reading it! (Which I did in a couple of days...)


This PDP-11 system had 48KB of core memory, a single RF11 disk, a TU56 dual DECtape drive and about six Teletype Model 33's were attached. (It might have had a PC05 as well; I don't recall exactly.) At then end of my first year, we were excited to learn that we'd gotten an extra 8KB of memory for it (not sure if it was an MM11-E or an MM11-F), which reduced some of the space crunch - on main memory, at any rate.

With the single 512KB RF11 disk (which suffered a failure at least once), each student was blessed with a quota of 5 blocks of disk, total! I conned the math teacher into mounting a DECtape for me as extra storage but alas the drive didn't like be on and used constantly, so it died and that was the end of my 'limitless' file space!

I didn't write any interesting programs on it. I did take a programming course and I took all of the Teletype printouts of programs for that and taped them into a wire-bound notebook, which I still had recently (I can remember clearly what it looked like) but although I just spent a couple of hours looking for it, I couldn't find it (alas).

One program I recall writing there was for an electronics class; it 'graphed', on a Teletype, the predicted voltage decay in a RC circuit. I took an electronic switch and built the matching actual circuit and compared the model output with what actually happened.

DEC had unwisely made mention of the SYS() call (in the explanation of one of the error messages, IIRC) in the Basic-Plus manual (which anyone could get) and the student from whom I had borrowed the manual from had done some 'exploring' in how to use it. Although I see that all the 'interesting' capabilities were restricted to privileged accounts, perhaps that was a later restriction. We also knew about PEEK().

In conclusion: I love PDP-11s, finest computer ever made. RSTS-11 was my intro to them. I now create/maintain the PDP-11 content on the Computer History wiki: https://gunkies.org/wiki/PDP-11

# Norman Long, British Aerospace, DEC Software Services

The delivery of the 6th PDP-11/20 S/N 230 in the UK to Hawker Siddeley in Kingston, Surrey the home of the Hurricane, Hunter and Harrier aircraft in August 1970 was a memorable occasion. It was the magic that not only spurred the enthusiasm of myself and my boss Richard Cannon but also changed my career, life and the spread of PDP-11s and VAXes throughout British Aerospace.

The machine was installed by two PDP-15 engineers from the Ealing Office, one being Nick Carr who I later got to know when I joined DEC and later transferred to VMS software engineering, Spitbrook Road, Nashua, New Hampshire. Richard (my boss) and I initially were like children and often would conflict on getting access to the PDP-11, sometimes on Sunday mornings and would get quite upset if one of us beat the other to the machine first, after all Sunday is not a normal working day. The first PDP-11 training course I attended was run in the CO-OP building near Fountain House by Eric Coates.

In those early days, DEC promoted a Paper Tape Software system as shown in the original PDP11-40 manual:

> "PDP-11 users needing an interactive conversational language can use BASIC which can be run on the paper tape software system with only 4,096 words of core memory. A multi-user extension of BASIC is available so up to eight users can access a PDP-11 with only 8K of core."

Initially we didn't have RSTS, we just had a paper tape system but once 8 User Basic was available, added extra terminals, one being in our design office probably1,000 yards away over a long line. Previously our design office had used a Ferranti Pegasus valve computer but once the IT department got an ICL1903, Pegasus had to go and designers were forced to use batch processing with 1 day usually failed runs. Once the design office got access to our PDP-11 8 User Basic system they were over the moon as they could get improved interactive computing again and later uses this as justification to get a PDP-11 of their own.

After expanding memory, adding an RF11 512KB fixed head swapping disk and DECtape our system would support RSTS and Peter Hoskins our brilliant Ealing office salesman arranged for RSTS to be installed. We were very excited about the prospect and when the day came, Rosemary Philimore, a very attractive blonde software specialist, came to install it for us. We were all mesmerized by her, including all the other males in the office…

*Editor's note: This was, after all, the 1970s! Rosemary was equally respected for her software skills. She subsequently joined Systime.*

During this period we were very well looked after by DEC Ealing Field Service engineers Alex Taylor and Iain Smith. Iain later became DEC UK Field Service Director based at DECpark in Reading who I met from time to time in "The Street" when I worked in DECpark.

In 1978 I joined DEC Ealing then Epsom office as a software specialist mainly supporting RSTS/E. I did allot of consultancy mainly helping Banking and Broking companies optimise the performance of their applications and systems. The main problem in those days was program size limitations which usually resulted in having to split applications into a main application and a file server application. One card account company had problems with being able to load a large RMS indexed file database every night for use the next day and using the normal RMS add record method was too slow. The solution was for me to write a RMS indexed fast file load utility in Basic-Plus-2 under RSTS/E which built the file from the data level up adding each index level on top of one another one at a time. I later moved on to VMS like most DEC software specialists did. By the mid to late 1980's DEC was finding itself in competition with IBM and software specialists were expected to support sales people more in Pre-Sales activity rather than the previous mainly technical support role. At that stage after eight years in Software Services I was fortunate to be able to transfer to a role in the ALLIN1 development team, Software Engineering, Reading which took me more back to my roots.

Happy 50th Birthday RSTS, a truly remarkable system which very much lived up to digital's reputation for quality and superb engineering.!!!

## Paul Laba, RSTS Development Manager

I started my DEC career at MK01 in Merrimack, NH on Jan 2, 1982. I'd given up my cushy job as Director of Computer Services at Le Moyne College (a small Jesuit college in Syracuse, NY, where I developed administrative software for the college on our trusty PDP-11/45), to join the ranks of a "real" engineering team, developing and supporting RSTS/E, the operating system I had come to know and love while studying Computer Science at Le Moyne in the late 60's.

I spent that first morning in boring HR orientation meetings, filling out an endless stream of forms and learning how to behave myself in public (my notes on the subject are sketchy but it does say something about no alcohol on the premises). I then went upstairs to meet my fellow RSTS workers and my new supervisor, Dick Goodwin. Dick owned the CUSPs (anyone remember CUSPs? All that Basic-Plus code?). Dick brought me down to the cafeteria, bought me lunch and outlined what I'd be working on ("Sysgen? Really?"), then said, "Come with me, we have to get ready for our group meeting at 1 pm." I dutifully followed him out the door and through the parking lot, to his 1970 VW Microbus, where he opened the back, opened an ice-filled cooler and handed me a cold beer. "Here, you're gonna need this," he said.

"Ok," I thought to myself, "this job is gonna be something."


And for the next 16 years, it was certainly "something" for me working in the RSTS group. I can say with certainty that I never met a more talented or fun group of people at any other time in my career, before or after. The friendships I formed over those years have, in many cases, continued right up through today.

And that leads to my second story...

One day in the late 80's, as I was walking through the parking lot (yes, the same parking lot), on my way into the office, I passed three of my favorite RSTS engineers, Jack DelBalso, Vernon Miller and Brian Nadeau, sitting in Jack's 20-

year-old Ford station wagon. "What are you guys up to?" I asked. "We're going for White Castle sliders," Jack said. (Sliders, for those unfamiliar with the White Castle vernacular, are tiny hamburgers.) "Wanna come?" Vernon asked. "Um, it's 8:30, a bit early for sliders, isn't it?" "Nah," Vernon said, "we're gonna go buy a bunch of them so we can freeze them." Never one to pass up a food adventure with these guys, I said, "Sure, why not" and climbed into the back seat with Brian.

"So where's there a White Castle around here?" I asked as we headed south. "I didn't know there was one in Nashua." "There isn't," Vernon laughed, "the closest one we found is on the NJ border." "What?? We're driving to New Jersey?" "Don't worry," Jack said, "we'll have you back by dinner time." And so, yes, we spent the rest of the day driving from Merrimack NH to a White Castle joint 20 miles from New York City, where we loaded up on boxes of sliders, gallon-sized jars of pickles and anything else we could convince the incredulous manager to sell us ("You drove here from New Hampshire? Why?").

When I arrived home around 7 pm that evening, my wife asked (as usual), "So, how was your day?" "Oh, you know, the usual." "But I do have this box of frozen White Castle sliders and jar of pickles that we'll need to make room for in the freezer." "Where'd you get those?" Karen asked, "I didn't know there was any White Castles around here." "Uh..."

Tragically, we lost both Jack and Vernon to cancer. I have Jack to thank for getting me into DEC and the RSTS group back in 1982 and Vernon for a wonderful friendship that lasted through the end of his life in 2016.

There were so many of those "OMG" days back then, (I certainly could have used that acronym back then), many I'm sure I've long forgotten. Mind you, I do clearly remember was the endless hours we spent debating stuff that really didn't matter! They were surely happy times for me.

By the way, someone had mentioned the "Bozo the Clown" DECUS story in their memories. Just to set the record straight, it was *not* me that dressed as Bozo! It was Jim Carey, a talented DECnet engineer in the group, attending his first DECUS. Jim and I started at DEC on the very same day (Jan 2, 1982) and we stayed good friends throughout our years in the group. Jim had a love for the absurd and introduced me to all things Monty Python. I'm not sure who came up with the idea of renting the Bozo costume for the follow-up panel discussion but it was brilliant and a classic.

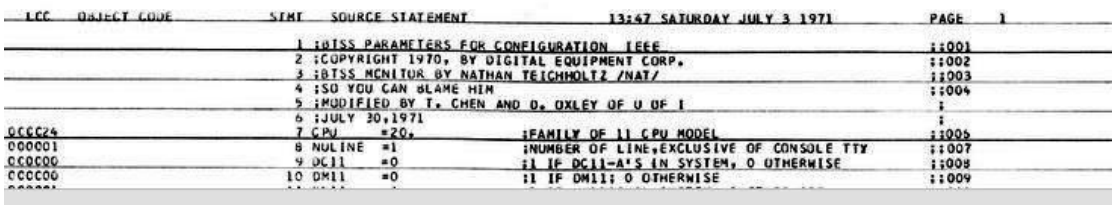
But to be clear: Underneath all our fun and antics was a group of very talented engineers, engineers who truly loved their work, loved writing code and loved working together to build a better RSTS operating system for their customers. It was and still is, the perfect combination of culture and work ethic for success.

# Paul Koning, RSTS Development Team

I started on RSTS with V4A-12 in the summer of 1973, at Lawrence University as a student. Somehow, the university got a RSTS source kit, which they printed (at 30 cps on their "fast" terminals). And somehow, I talked the computer center boss (Michael A. Hall) into letting me take those printouts with me over Christmas break. When I came back for the spring term, he hired me as a student programmer; that was the start of my professional programming career. A couple of years later he commented to my father *"I make it a practice to hire talented young students BEFORE they can become dangerous"*. During my second year there, Anton Chernoff was the educational software chief; he came there from DEC (RT-11) and went back to DEC the year after. He took me under his wing as one of his student staff.

Not long after I was hired, we encountered a DECtape bug, very obvious because our DECtapes were the primary user file storage medium. On April 7, 1974, all the new DECtape files showed dates of 01-Jan-70. DOS had defined the DECtape date field as 12 bits, so it wrapped at 4096 and RSTS-11 had faithfully copied that mistake. I found the error and created a patch for it; one line of course, changing that field to 15 bits. We sent it to DEC which published it as an official RSTS patch.

After Lawrence I attended the University of Illinois, where I did some PDP-11 work but mostly worked with other machines. But while there, in 1976, I discovered something very strange: an old assembler listing from 1971 sitting on some pile of paper. It said it was *"BTSS monitor by Nathan Teichholtz /NAT/ so you can blame him"* -- modified by two people at the University of Illinois. I still have that listing; here is the start of it:



```
LOC   OBJECT CODE      STMT   SOURCE STATEMENT          13:47 SATURDAY JULY 3 1971          PAGE   1

                         1 ;BTSS PARAMETERS FOR CONFIGURATION  IEEE                        ;;001
                         2 ;COPYRIGHT 1970, BY DIGITAL EQUIPMENT CORP.                     ;;002
                         3 ;BTSS MONITOR BY NATHAN TEICHHOLTZ /NAT/                        ;;003
                         4 ;SO YOU CAN BLAME HIM                                           ;;004
                         5 ;MODIFIED BY T. CHEN AND D. OXLEY OF U OF I                     ;
                         6 ;JULY 30,1971                                                   ;
000024                   7 CPU     =20.          ;FAMILY OF 11 CPU MODEL                   ;;005
000001                   8 NULINE  =1            ;NUMBER OF LINE,EXCLUSIVE OF CONSOLE TTY  ;;007
000000                   9 DC11    =0            ;1 IF DC11-A'S IN SYSTEM, 0 OTHERWISE     ;;008
000000                  10 DM11    =0            ;1 IF DM11; 0 OTHERWISE                   ;;009
```

It is a very early RSTS, with Basic-Plus stripped out and a binary program loader put in its place. With sufficient care, that would allow you to run assembly language or compiled programs. And indeed it worked at least somewhat; I also have a foot or two of Teletype console printout showing it.

While there, I tried to get a summer job at DEC. Anton Chernoff, by then in RSTS/E development working on V5B, invited me to stay with him but told me he was putting me to work: he needed PIP.SAV for the V5B SYSGEN changes. I said *"ok, I can hack up something primitive"*. He told me no, he wanted a good and real one. I set to work and by week's end I had a working version that could

copy files, about 100 times as fast as the regular (Basic-Plus) PIP. This was the start of the PIP we still know and love; to this day my initials are first on the list of authors. That was almost 3 years before I became a DEC employee.

I joined DEC in 1978, working on support for Typeset-11 (which runs on RSX-11/D and IAS). In 1980 I transferred to the DECnet/E project, to work as junior team member for the lead developer, Rich Witek. About 2 months later he left to join the DEC chip development team, leaving me suddenly as the lead designer for the DECnet routing layer. That meant I had to learn to read DECnet architecture specifications and how to translate them into running code. My networking career started at that point, which I continued to do full time for another 21 years and as part of my work to this very day.

I still use RSTS, for fun. I still have the Pro-380 to which I ported RSTS and I hope to make that available to others at some point. For that matter, I also tinker with even older software -- CDC mainframe software I worked on at the University of Illinois and software for peculiar Dutch computers (Electrologica) on which I wrote my first programs in 1969 and the older of which is only 2 years younger than I am.

## Peter Clarke, Group IT Manager Jackson Building Centres

The business was first formed as Jacksons Hardware Ltd in 1946 by Eric Jackson. Whilst its roots were in the retail market, the business quickly expanded into the “trade” and out of Lincoln.

In the early 1950s the first branch was opened in Spilsby. Newark was opened in the late 1950s and traded from large premises offering a full range of materials, including timber and tool hire. Grantham followed in the early 1960s again offering a full range of materials, including timber and tool hire.

In 1968 Jacksons Hardware took over Shipley and Co and became Jackson Shipley Ltd. This meant a second business in Lincoln and a new branch in Gainsborough. The two Lincoln businesses were consolidated into the current Pelham House Head Office in 1971. The company grew and by the middle 1970s had about ten branches: this subsequently grew to 30 branches.

In 1983, when we moved into ex-Robeys buildings, a PDP-11/70 with 3 * RM03s (67MB) running RSTS/E was installed. Initially on a ground floor but subsequently on a first floor via a crane and blocked off roads.

At the time it was envisaged that this one machine would run all the Jackson business. The first elements of the business to be computerised were the Product File and Customer File, with an objective of being able to Price and

Invoice customer transactions, which at the time were manually keyed in. All application programming was done in Basic-Plus.

During the next few years the Product File, Customer File, Customer terms, Credit Management and Purchase ledger were all built. This was swiftly followed by the introduction of a POS (Point Of Sale) system, which was introduced at Lincoln branch first, and saw the company trading with screens on the counters for the first time. DEC VT100 devices had been upgraded to the new DEC VT220s and these were used on counters along with OKI 182 printers to produce customer documents.

The desire to start computerising the branches meant recognising that we couldn't do that using the PDP-11/70 solely, as comms speeds and other limitations meant that we could not run remote terminals at branches sensibly.

So the company then started down the road of introducing PDP-11/83s at all branches, transferring sales data to the 11/70 each night for invoicing etc and receiving updates on products, prices, customers, terms etc in return.

Initially this was all done over 2400 baud modems, which were replaced by 9600 baud modems, moving to ISDN lines and eventually Kilostream lines as the demands on data transfers grew. The advent of Kilostream lines allowed us the offer much wider, real time visibility to data on all systems.

Establishing a stable and secure model for this allowed the company to grow easily, without the IT systems holding back growth. Adding another branch system, with a standalone 11/83 in the branch and configuring this in the PDP-11/70 'Head Office' system become was a well-drilled and practised solution.

Eventually the PDP-11/70 replaced by DEC Prioris 6200 with multiple Intel processors, running SCO OpenServer (Unix). Each PDP 11/83 at the branches was replaced by a DEC Server 590, Intel Pentium chip based systems, again running SCO OpenServer.

At the turn of the century we completely centralised the systems onto two IBM systems, in separate building, with mirrored disc arrays, with the servers alternating being the live server to ensure a fully automatic and resilient DR (Disaster Recovery) solution. Both IBM systems were Netfinity 8500 with 8 * Intel Xeon CPUs, 3Gb of memory and 16 SCSI disk drives configured as RAID 0+1. Each system could support 1,000 online users. Showing a complete lack of professionalism, the Silver supplier named these two systems as Jack and Jill. These two systems ran for over ten years.

Using RSTS/E systems with Basic-Plus allowed the company to grow relatively easily, without the IT systems holding back growth and at a low cost per user.

We have recently moved onto a new platform and eliminated our Basic-Plus programs, some of which had been run for 30+ years.

## Peter Dick, Silver Programs UK

Purveyor of Fine Software since 1970

The 4120 in operation at Hartree House. Team leader Mick Payne is on the right and computer assistant Peter Dick is at the rear.

Having spent the previous three years working on mainframes, such as the NCR Elliott 4120 (which was a tape based system without any disks) shown above, on seeing my first PDP-11 RSTS system at Allen Computers (London) in January 1974, I asked “But where is the computer?”

It was quite a culture shock to discover RSTS catered for multiple users including programmers; online updating… and all without a single a “computer operator” in sight. And the hardware only occupied a couple of 6ft high racks.

Terry Williams of Blake Computer Services was my boss: there had been two applicants for the job, myself and Chris Blake. I only got offered the job as I didn’t bring a dog to the interview.

My first SYSGEN for RSTS-11 V4a took three days, most of the time spent OLDing the CUSPS from paper tape via a Friden Flexowriter at 10 c.p.s.

Terry reminds me of the challenges of computerising all aspects of J.M.Hill and Sons, a large building company, with only 4 x RK05s (2.4MB hard disk drives) and 16KB of user memory. One answer was to employ our own version of overlaying and split logical programs into pieces. The other answer was to add more memory and upgrade to RSTS/E. This was done before the first application, payroll, went life on Saturday 6th April 1974. This was successful thanks to the help given by the DPM (Data Processing Manager), Barry Isaacs.

By the late 1970s I had written a Report Generator, using table-driving techniques extolled by Terry, which included a speedometer. I thought reading at 1,000 blocks (1/2 megabyte) per second was as fast as any computer would ever achieve… There again, I didn't think I would ever see anything faster than 9,600 baud for remote access via a phone line.

Let's not forget pre RUN $BACKUP when we used ROLLIN, PRESRV and SAVRES while in the days before the Internet we had … RSTS Professional and DECUser magazines, The Small Buffer, Software Despatch, DECUS biannual Conferences, SIGs, LUGs, DECUScope, DEXPO, COMDEX, CeBit, DECworld, Kermit and a box of null modems / gender-benders.

Then there were the DECUS T-shirts / button badges:

> Digital Has IT Now, But We Make IT Work
> Please Say HELLO
> This page intentionally left blank
> My MT0: is in Shoe Shine Mode
> I'm a Twit Biddler
> Escape from TECO
> There are 10 types of people in the world…
> … those that understand binary and those that don't

It is fun looking back at the cost of hardware: in 1976 I bought a memory typewriter for £1,800 and then in 1983 I purchased my first PDP-11/73 system for £10,000 from Darkcrest. Not just because Nick was a super salesman but because Roger Box was their technical director. And he was brilliant.

I made my first DECUS USA presentation in May 1985, celebrating the RSTS 15th birthday. That was also the year Silver opened a front (I mean an office) in USA run by the amazing Billy Maxman, who prime job was to process all the US$ checks (cheques in English).

In May 1990, I was honoured to be the Master of Ceremonies for The RSTS 20th Birthday Celebration: my sole qualification being I spoke English while everyone else spoke American.

Pub trivia facts:

In 1983 I formed The BASIC Party and stood for Parliament in the UK General Election. I didn't win.

I was the Lead name in a class action court case against Microsoft. After seven years they settled.

I have been signing off emails for over twenty years as "Bye/P" in homage to BYE/Y, BYE/F or BYE/I but nobody has ever appreciated this gesture.

My highest income per character typed (£30 per character) was for entering "RESET" at RSTS boot time, so that the memory installed 18 months earlier would actually be used…

I personally designed the button badge that compared RSTS/E with VMS: BEAUT/E and the BEAST

In the 1990s, as PDP-11 hardware development suffered from "VAX is the answer, what is the question?" we focused on providing more power for RSTS Users. We offered various solutions, including OMSI SORT1, BAS24K, BP3, The Link and Quickware Engineering's QED 11/95 products.

Later on, when DEC formed Digital SME, we were appointed Value Added Resellers of applicationDEC systems running SCO Unix V3.2R4.2 on Intel 486 chips. We also bought the rights to Trans_Basic. This combination allowed applications, without any changes to Basic-Plus source code, to run exactly as they had on PDP-11 hardware … but about fifty times faster!

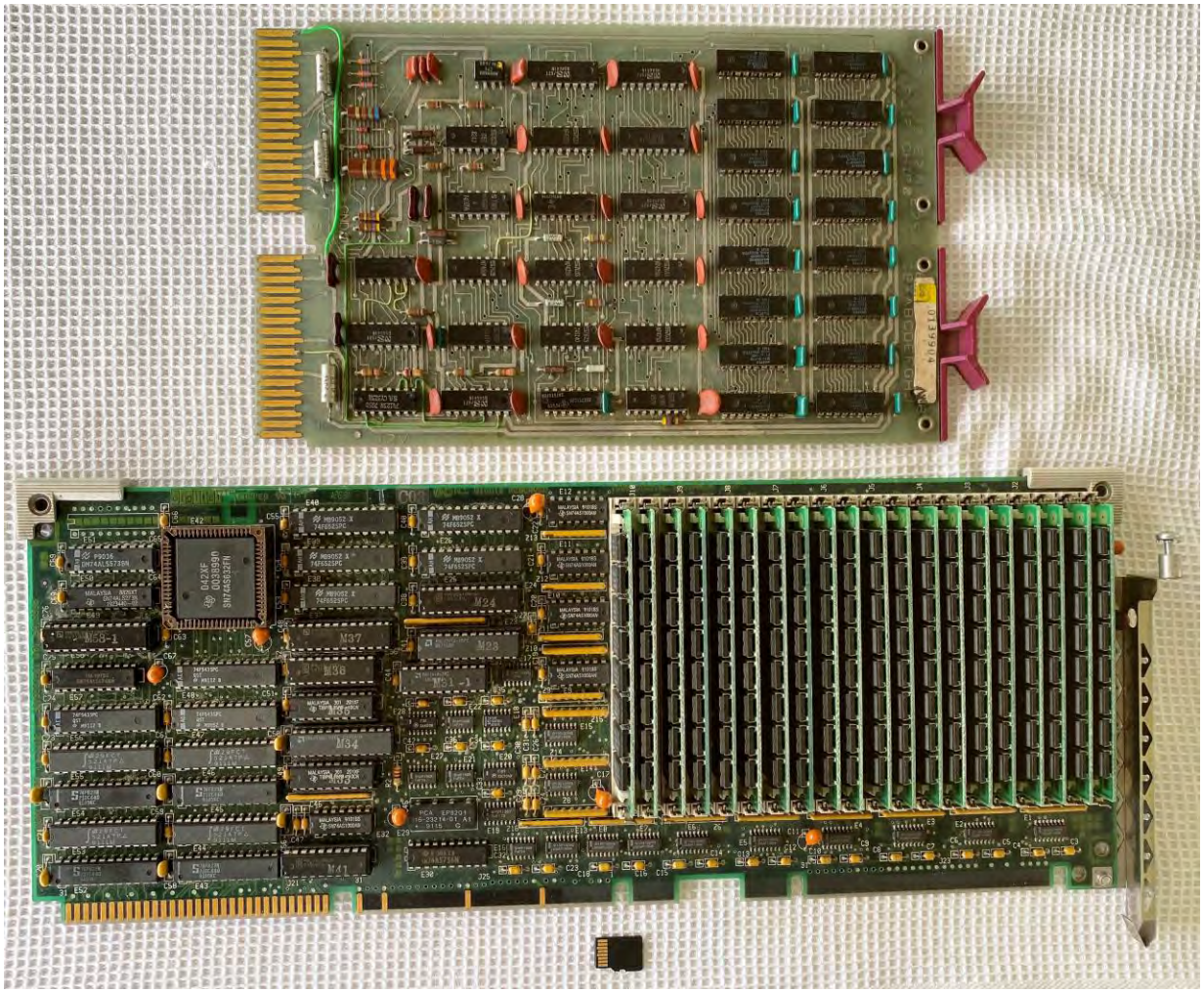
In the run-up to Christmas last year (2019) I was invited by Hans Schilders to speak at the handing over of thousands of Chromebooks, provide by the Rotary Club, to local (Dominican) schools. I wanted to explain how modern computers are a million times faster/bigger than my first PDP-11.

For the first example, I used "time" as an example: There are twelve Days of Christmas. A thousand times faster … about 16 minutes and a million times faster … one second.

My second example showed the falling cost per megabyte in the last column:

| | | | |
|---|---|---|---|
| 1975 DEC PDP-11 M7944 | 4 KW | £500 | £64,000 |
| 1991 applicationDEC 433MP | 16 Mb | £2,500 | £ 156 |
| 2019 SanDisk Micro SD Card | 64 GB | £10 | 1/100 of 1p |

I have these three examples of hardware with me here in Dominica but I suspect the nearest PDP-11 or applicationDEC machine is a few miles away…



I wrote the RSTS 80th Birthday document after someone at a DECUS RSTS USA SIG meeting got excited about the then approaching RSTS eighteenth birthday and said eightieth by mistake. So I nicked the idea.

In case you haven't already guessed, I am also the Editor of this document, once again benefitting from other people's work. The story of my life…

Looking back, supporting PDP-11s, RSTS/E and Basic-Plus provided me with a wonderful living for 28 years. But I was just the organ grinder, doing nothing more than slowly rotating my arm, while my more talented suppliers were the amazingly monkeys doing the tricks. On the other hand, I held the tin that collected the money… Many, many thanks to everyone…

RSTS 80th Birthday Document
http://www.silverware.co.uk/rsts_80th_birthday.htm

RSTS/E Optimisation Seminar Documentation (1 whole day!)
http://www.rsts.org/~pdick/corner/RSTS_Optim.php

# Phil Traviss, British Steel

## Background

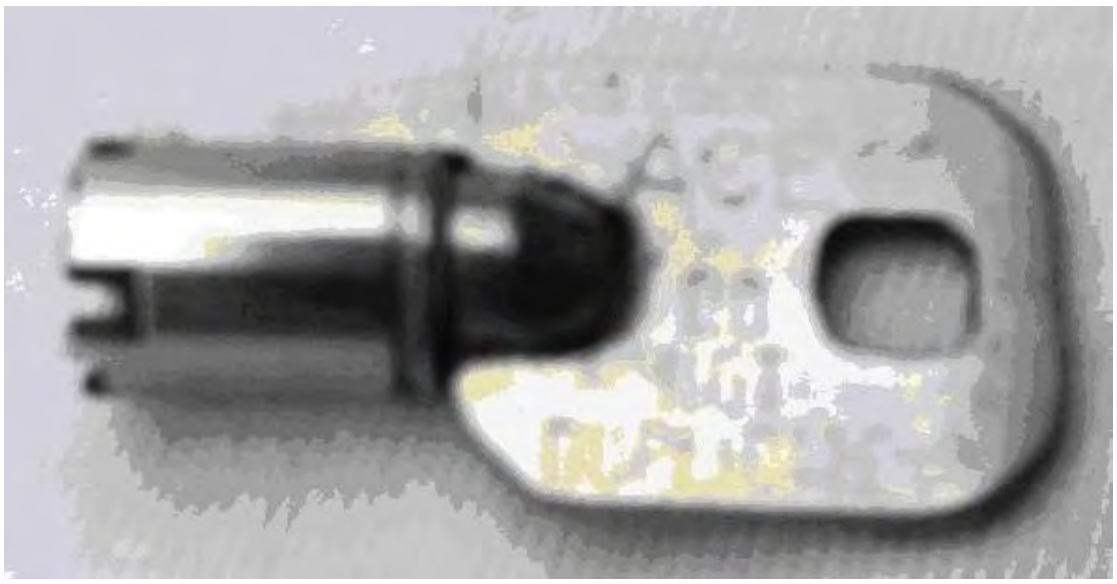
I first came across RSTS/E as a student at Loughborough University in the 1970's. My first job after graduating was as an applications/systems programmer at British Steel Teesside Laboratories, where they also ran the same system. I was given account [15,12], which I only remember as it was <C/R> <L/F> in octal. We used RSTS/E until it was replaced by VMS running on DEC VAX systems, after which I became a migration specialist.

## Mementoes

My RSTS Professional T-Shirt.  I'm not sure how this survived but there it was in the bottom of the drawer.


## Console key

Should perhaps have handed it back, but, well, you never know when you might need it…



Exchangeable disk pack handle.

I had to forcibly remove this memento from an (already failed) disk pack lid to release it. I had an idea that it would make a very cool gear lever if ever I owned an automatic car (I think I had a mini at the time).


Memories

Why don’t you just use NXTVAL()?

The first job I was assigned involved reading in a text file containing numeric data. After a couple of days of programming effort (dealing with converting strings to their numeric equivalents), it was only then revealed to me that several home-grown (and under-documented) enhancements had been added to Basic-Plus by the British Steel team, one of which was NXTVAL() which basically replaced all the work I had done so far.

Where’s the Yellow Frog?

The majority of the dumb terminals we had around the labs were wired via 20mA current loop, terminated with a 5-pin DIN socket. Steve Gray designed a DIN plug with 2 LEDs to check for line problems. Since LEDs are generally 20mA, no other circuitry was required. The plug was wrapped in yellow electrician's tape and the two 'eyes' stuck out the end, hence yellow frog.

### Head Crash!

Once in a while an embarrassing/expensive pile of damaged disks would appear outside the computer room door (RP05's?). To prolong the suffering, the DEC engineers had to boot up the 11/70 in diagnostic mode to perform calibrations after replacing the heads, meaning the whole labs had no service. I tasked myself to write an online version of the DEC diagnostics (e.g. repeatedly seek 20 cylinders, etc.) that mimicked the functions they required, so that we could let the users on while the DEC engineers were working on the disk drive.

### You are in a maze of twisty passages, all alike.

I'd be extremely surprised if no other submissions mentioned the infamous 'ADVENT' game. Mapping the maze was one of the key aspects of progressing through the game, and a lot of the fun (and tropes) of trying to solve the puzzles is still fresh in my mind even now.

### In Conclusion

RSTS/E brought together like-minded people and encouraged elegant solutions that you could be proud to share with colleagues. It wouldn't have been as rewarding if there were no others to share in the fun.

## Pierre Rügländer, DataUnit, Switzerland

50 Years, half a century.

My life changed after contacting the RSTS Development Team in the USA and being told that Peter Dick had the answers to all my problems. Or was he the problem? My English was never perfekt.

So, I ended up buying software licences for DEC RSTS systems… and BAS24K… and BP3… and The Link … and Trans_Basic. And I attended various DECUS USA symposiums where I met many of the RSTS development team.

However, most of my memories are about project estimates: two full days allowed for The Link installation that took only ½ hour.

At the other extreme, I remember Peter spending an entire afternoon trying to explain the use of SPEC% to change an explicit disk block lock into an implicit disk block lock to my Swiss programmers who only spoke BASIC English.

I would also point out good memories overrule all silly things written on boxes…

My real claim to fame is I videoed The RSTS 20th Birthday Party in New Orleans in May 1990. You can see the highlights on YouTube.

https://www.youtube.com/watch?v=LK3-kHhfoMc&t=28s

Small world: my RSTS 20th Birthday video was uploaded onto YouTube by Peter's Goddaughter Michelle, daughter of David Allen of Allen Computers.

I am delighted to say the RSTS/E SIG (Senior Interest Group) continues to meet: this snap was taken September 2018. Left to right: Myself, Tim Chamberlain, Trevor White, Peter Dick and Peter Clarke.

Finally, this is what spontaneously came to my mind.

This is a **R**eally **S**uperb **T**hing **S**adly **R**oyals, **S**oldiers, **T**axidrivers, **S**ingers don't care but we do remember - Happy Birthday RSTS.

## Richard Angelini, Bally Systems, Reno Nevada, USA

RSTS/E - I cut my programming teeth on this great operating system and Basic-Plus in the early 80s.

I visited the Digital Equipment Corporation (DEC) RSTS/E offices in Maynard Massachusetts in order to get my first set of RSTS/E user manuals. Those were exciting times in multi-user computing.

Over the ensuing 30 years I earned my living from that operating system and Basic-Plus. We (Bally Systems) had well over 50 casino sites running RSTS/E until DEC's last RSTS/E 10.1 release in 1992 and around 150 sites running Basic-Plus source code in 2010 albeit via Trans_Basic !!! Today (June 2020) there are a handful 20-25 casino sites still running that code.

All hail RSTS/E !!!

## Richard Cannon, British Aerospace

While at Hawker Aircraft (later part of BAES) in early 1970 I ordered a PDP-11.

With Norman Long I went to the Electronics and Instrumentation Show. The first PDP-11 in the UK arrived that day and we were there long after closing trying to get a program we had written to run; we succeeded in running the first PDP-11 program written in the UK.

We received PDP-11/20 Serial Number 230, the 6th one in the UK. I think it cost £5,400 with 8kb of core memory (I think) and a teletype; we added an extra 8KB of RAM which cost £1,960. The PDP-11 and VAX11 became the standard mini-computers used by BAES.

My career changed from Mechanical/Electrical Engineering to Computing. After some years Norman moved to work at DEC. At 85 I am still very involved with PCs.

## Richard Foulds, Leckhampton Computers

In the mid 1980's, we took on a job for a client, Bartletts in Leeds, whose previous software company had deserted them mid project.

It was an insurance brokerage running RSTS on a PDP 11/44 with their Basic-Plus applications integrating with a word processor (WPS?) and to meet

deadlines I worked there 100+ hours a week and pulled many all-night stints, alone, in the big old building.

Note the “new” extension that was added sometime during the last 40 years…

After a one such night, I was leaving and heard one of a group of decorators, also working in the building, telling his boss that he wouldn’t go back inside because it was haunted and that he had seen a grey headless woman walking down the top floor corridors.

He was genuinely scared.

I mentioned this to the IT manager the following day and she closed the door and asked me to keep quiet because she didn’t want the rest of the staff to know.

Apparently this figure had been seen many times over the years and was a woman who had been murdered there 150 years before.

Nights there were never the same after that!!!!

Suffice to say we completed the project in time and the company remained a client until being taken over many years later.

## Richard Gutteridge, FBCS

In the early 1970s I was working as a programmer for United Kingdom Chemical Information Services (UKCIS) based at the University of Nottingham. I was working on ICL and IBM mainframes and had become accustomed to a 2 or 3 day wait for my program tests to be returned, only to find that the compilation had failed.  One of the academics lecturing in Chemical Engineering was a certain Dr George Coggan and he convinced me to go and work for him in a company he was starting called Gamma Associates.

George was in the process of obtaining a PDP-11 running RSTS/E and his concept was to sell timesharing services to businesses. My job was to manage the bureau and design systems and programs using Basic-Plus. When the system arrived the effect on me as a developer was amazing and my personal productivity went through the roof. Compared with using machine code or COBOL, Basic-Plus was just incredible and interactive debugging opened up a brand-new world.

One of the niches we found was providing a service for Newspaper and Magazine Wholesalers around the country. Eventually we had to work shifts to keep up with demand. Once per week each wholesaler was sending us a load of paper tape transactions, these we duly processed and two or three boxes of printed-paper were returned to the wholesaler. It seems unreal that the 2.4MB RK05 disc drives we used were removable… we had one disc for each customer.

The focus in Gamma gradually moved from selling time on our RSTS systems, to selling the systems themselves as an OEM. Later Tim Rhodes joined Gamma to work on software to monitor investment portfolios and I think you know Tim.

I have the greatest of pleasure in wishing RSTS/E a very happy 50th and thanks for the memories!

## Robert Schaffrath, New York High School, USA

My first contact with RSTS/E was in a 10th grade introduction to BASIC programming class back in the fall of 1977.

My high school, along with most others in the county, rented time on a PDP-11/70 from the county's BOCES (Board of Cooperative Education Services) system (Nassau County, New York). We had two 300 baud acoustic couplers with LA36 terminals as well as two ASR33 teletypes. We could do punch tape on the ASR33's while waiting for an LA36 to become available at which point we would downgrade the speed to 110 and hook up the ASR33 in order to send the tape.

I still remember our class account was [145,101] and the password was OMBX.

We had a whopping 25 blocks of disk storage. The version of RSTS/E running at the time was V06B-02. Sometime over the winter break, the BOCES center upgraded the system to V06C-03 at which time we got TECO as well as control-R and control-T functions (apparently our system was somewhat special as I have a V06C-03 system running with SiMH and control-T is not supported anywhere in the distribution. The TTDRV source code has the mini-SYSTAT functionality shown but none of the shipped system libraries have support for it).

When I took the advanced programming class, we learned additional Basic-Plus programming as well as FORTRAN (which served me well when I went to college and had to take a class in FORTRAN programming which I already knew). We also learned some rudimentary TECO editing.

After graduating college, my first brief job was Basic-Plus programming under RSTS/E V9.0 at a tiny shop running it on a PDP-11/73. We also had a "production" PDP-11/34a running V8.0 but I had very little interaction with it in the short time I was at the firm. Others moved code from the development system to the production system.

After that job, that was my last contact with RSTS/E until I built my first SiMH system running 7.0. Later on I set up versions of V9 from 9.0 to 9.6 and ultimately 10.1. V9 was rather annoying because I enjoyed using PEEK to access various system data structures and as V9 progressed from V9.0 to V9.6, a lot of the PEEK functionality went away due to those structures being moved to upper memory. DEC did give ample warning with the V9.0 release that they would be removing functionality in forthcoming releases but it was annoying having to use FIP to get access to data indirectly rather than go right to memory with PEEK. It was only in the past two years I finally got a V06C system running when the defective distribution that had been floating around on the Internet for years was fixed by a clever enthusiast.

## Roy Sharp, Gamma Associates UK

I was introduced to RSTS/E in 1975 when I went to work for Gamma Associates in Nottingham. I'd started work 18 months earlier at UKCIS on the Nottingham University campus but was made redundant when they realised that they had overstaffed. Moving from COBOL coding sheets and punched cards on an ICL System 4 to the immediacy and flexibility of Basic-Plus and RSTS/E on a PDP-11/44 was exciting. As employee number 13 there was a lot of learning to do, quickly.

Once you got used to tiny programs, short variable names and as few line numbers as was possible while still leaving the code readable (I think they saved 8 bytes each!), it was amazing what could be done.

Gamma provided a bureau service for wholesale newsagents, so it was a 24-hour operation some days. I remember overnight processing and print runs, deleting work files off hard drives before the running program crashed due to low space. Hacking the code to restart the process when I was too slow.

It was also expected that when you called out the engineer you would be able to tell him where the problem was – tape drive, memory card, etc. of which RSTS/E was quite capable. I also recall that RSTS/E was great in a crisis, if there was a

bad block on a magnetic tape or disk, with a little code you could read everything before and after the damaged area so rarely lost everything.

Fun times indeed! Don't miss modems running at 10 cps though.

## Russ Ristine, Bally Systems Client and Consultant

My best memory of RSTS (and Basic-Plus) was a response I received to an article on Resorts International's casino management system application in DEC Professional (January 1987), in which I expressed concern with the 16K-word maximum program size under Basic-Plus, describing how we had to constantly work to streamline the code.

Several weeks later, I received a gift in the mail of a coffee cup ("Old Programmers Never Die, They Just Decompile") containing a note from someone called Peter Dick telling me of BAS24K, saying "Problem Solved".

With that, we entered into the care of the team at Silver and never looked back. Thanks for the memories.

*Editor's note: as at June 2020, around 25 large casinos are still running original Basic-Plus programs written by Bally Systems, now under TBX. Some of these programs were written 43 years ago…*

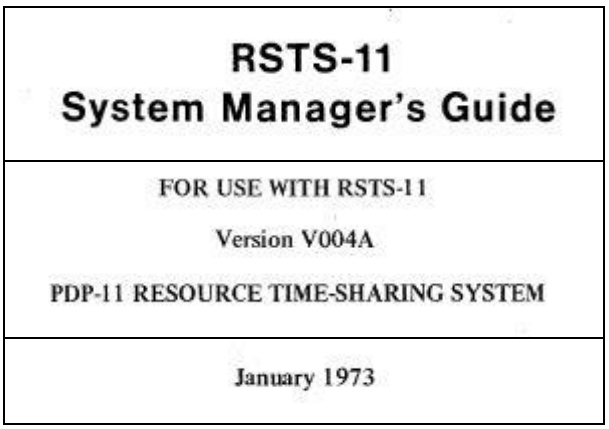
## Simon Maltby, Time-Line Computer Archive, UK


The above picture shows the recently donated Jackson's 11/70 … in need of some work! The three terminals (temporary location) on top of the main cabinets are, left to right, VT05, VT100 and VT50.

Historically DEC created buttons/badges for each new release, including two classics (in those days):

RSTS 6C is Sexy
Share RSTS/E with a friend

There are several sites that have scanned copies of old RSTS manuals including http://bitsavers.org/pdf/dec/pdp11/rsts/



**RSTS-11**
**System Manager's Guide**

FOR USE WITH RSTS-11

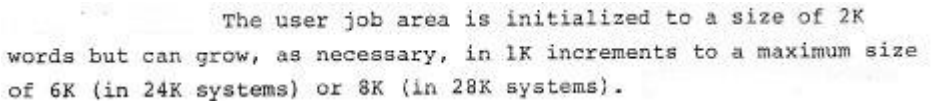
Version V004A

PDP-11 RESOURCE TIME-SHARING SYSTEM

January 1973

Note the above product is called RSTS-11, the abbreviations should have been RTSS which stands for Resource Time Sharing System.

Reading through these manuals confirms RSTS has come a long way since those early days where the original maximum program size was quite small


The user job area is initialized to a size of 2K words but can grow, as necessary, in 1K increments to a maximum size of 6K (in 24K systems) or 8K (in 28K systems).

Another source of manuals can be found at the UK Centre for Computing History in Cambridge: e.g. the 1980 RSTS/E Primer for Version 7.0
http://www.computinghistory.org.uk/det/19225/RSTS-E-Primer/

Great article on an early PDP-11/50 RSTS system
http://www.columbia.edu/cu/computinghistory/pdp11.html

And, of course, the opposite of leading edge technology…
http://pdp-11.trailing-edge.com/rsts11/

A website dedicated to preserving the history and legacy of the PDP-11
http://www.pdp11.org/

And here is another great website dedicated to PDP-11s
https://pdp-11.com/

Here are some examples from the Digital UK 1973 Price List (in UK pounds). Note in those days the exchange rate to the US dollar was roughly £1 = $2.2

```
VT05B-CA B&W Alphanumeric text CRT 20 lines x 72 cols   £ 1,440

MF11-LP 16K bytes parity memory                         £ 2,940

LP11-JA 300 LPM 132 column Upper Case only printer      £ 9,040

RP11-CE 40 megabyte RP03 removable disk drive inc ctrl  £16,400

QR430-AD RSTS/E 32 User Timesharing Licence on mag tape £ 1,760
```

Also famous in 1973… the first print of the book called `101 BASIC Computer Games` by David Ahl.

There are RSTS simulators on the web, for example

https://skn.noip.me/pdp11/pdp11.html

https://www.pcjs.org/machines/dec/pdp11/1170/panel/debugger/rsts/

http://elvira.stacken.kth.se/

Finally: you can find out more at our Time-Line Computer Archive website, which includes videos made by both Peter Dick and Tim Chamberlain.

https://t-lcarchive.org/

## Steve Gray, British Steel, Swift Tools

I'd been working at British Steel Research Labs in Teesside for a year or so, writing furnace-modelling software in Algol 60 but I reckon it was my "cribbage" program that clinched the deal when I was "recruited" by David Peel to join his computer department.

Not long afterwards, in 1974 our first PDP-11 arrived, with RSTS/E (we were told to pronounce it "RUSTUS"). Originally while we expected RSTS-11 running on an 11/40, the order was upgraded before delivery to an 11/45 and it came with RSTS/E

This was exciting stuff. No longer did I need to book machine time, or prepare offline edit tapes. I could sit at a terminal and write software all day.

I've always had an intense interest in how things work, so some of my earliest programs were a disassembler that produced Macro-11 source, along with a binary "diff" program and assembler, all written in Basic-Plus. Using a repeated process of disassembling, editing, reassembling and binary comparison, I managed to create a complete set of sources for RSTS/E and Basic-Plus – albeit without comments. I continued to track DEC's code changes as RSTS evolved.

In the code, I discovered that DEC had made provision for shared runtime systems, so that same year I wrote an RT-11 emulator that would run the "real" Macro-11 assembler. Although it was superseded a year or so later by DEC's official offering to support Fortran, it proved a useful stepping-stone for subsequent Basic-Plus enhancements.

In 1975, DEC's Peter Cook from Reading paid us a visit. He'd heard about improvements I'd made to Basic-Plus and realised I was surprisingly conversant with the code. When he asked how I got hold of the sources, he was astonished to hear that I'd reverse-engineered them. A few days later a small parcel arrived containing microfiche listings for RSTS/E V06A. I could hardly wait to get my hands on a fiche reader!

The following year a set of RK05 disks arrived, with machine-readable source code for V06B. Having the comments provided a fascinating insight into the minds of the developers, especially Evans, Griffiths and Hart for Basic-Plus.

An amusing comment I spotted in the keyboard handler for the DH11 multiplexer was: *"The following two instructions are dedicated to DEC Special Engineering without whose help they would have been completely unnecessary".*

After setting up my own software business, Computer Craft, in 1987 and encouraged by Peter Dick, I went on to develop BP3 – a program to convert Basic-Plus BAC files into optimised machine code. Although the compiled code ran up to ten times faster, there was no real scope for improving the speed of the run time libraries, so execution averaged out at about three times overall. Three was also a nice number to use in the name, since it positioned it as a natural successor to DEC's BP2.

Since compiled code is around twice the size of Basic-Plus's PPCODE, I implemented code paging, allowing programs to be considerably larger than the available memory space – up to 1MB which, in the days of 32KB programs was enormous.

I can remember when the first 16Kbit memory chips appeared, there was talk that 64kbits would be the final upper limit because the cells were becoming so small that background radiation in the ceramic substrate would knock out the electrons. I now have 32GB in my mobile phone, yet I still look back on RSTS/E with fondness.

## An inside view of BP3

*Editor's note: I have included this technical memory as I think it gives an insight into how the RSTS Development Team created an operating system that allowed us, the Users, to benefit from third party additions/products.*

BP3 converts the PPCODE contained in a Basic-Plus .BAC file into native PDP-11 machine code. It was written in Basic-Plus so that it could take advantage of its own compilation. BP3 compiles itself.

It was implemented as a two-pass process comprising two separate Basic-Plus programs communicating through a temporary intermediate file.

## Pass 1

This pass builds a table that accumulates the sizes of all data items, variables and constants. Once everything has been accounted for, it works out how to arrange them in memory and assigns run-time addresses to them.

It also performs "folding" of constants. This is a recursive process so, for example if a program contains:

```
A = 10 : B = A + 6
```
BP3 will compile:
```
A = 10 : B = 16
```

To do this it has to determine that no execution path exists that could change the value of A before it's assigned to B.

Folding includes almost every operation (including type conversions) that doesn't have side effects. So an expression such as:

```
A$ = "C" + CHR$(32.0*2%+1.5) + MID("POT", INSTR(1%,"DOG","G"), 1%)
```
Will compile as though it had been: `A$ = "CAT"`

Functions that can have side effects will not be folded, the most obvious example being the SYS() function. The argument string can be folded but the SYS() function itself must be executed only at run-time.

The first pass also generates intermediate code, from which machine code will be produced by the second pass. This intermediate code describes each operation (e.g. LOAD, ADD, STORE) and provides addresses for the variables and constants but it doesn't create any real code or attempt any code optimisation at that stage. In many ways it's similar to PPCODE but uses 16-bit words rather than bytes, so that links and addresses can be included easily. Each code block is padded to be big enough for the worst-case code that might be required to implement it.

## Pass 2

This pass generates machine code and produces the final executable image. The first job is to lay out the data areas of the program image using the table of sizes determined by pass 1.

The next step is to identify common sub-expressions. This maintains a list of virtual "registers" and the values they currently represent. Generally, these virtual registers will exceed the availability of real processor registers and some decisions have to be made about how to allocate them. The algorithm works to minimise the cost by: taking into account how many times a value will be needed in later code; how much code is needed to create the value; whether it's worth the cost of saving a register to temporary memory and reloading it later.

Of the eight PDP-11 registers, R6 is dedicated as the stack pointer and R7 as the program counter, which leaves six general-purpose registers. A further complication is that some instructions (e.g. division) demand a pair of registers and some (e.g. MUL) work differently depending on whether their register is odd or even. All this is taken into account by the register-costing algorithm.

The next step is code generation and local optimisation. As previously mentioned, the intermediate code allocates enough space for worst-case code generation. So, something like:

```
A% = A% + 1%
```

Will have enough code space for:

```
MOV A%, Reg          ; move the value of A% into a register
ADD #1, Reg          ; add one to the register
MOV Reg, A%          ; store the register in A%
```

Firstly, the optimiser will notice that ADD #1, Reg can be replaced with INC Reg. Then it will notice that the value is being saved to the same address it was loaded from, so the register is not needed. The final sequence will become:

```
INC A%               ; increment A%
```

This will be followed by four NOP instructions to (temporarily) fill the allocated space.

Code targets correspond to line numbers that are the object of a GOTO or GOSUB, or FN call but they can also be implicit, such as the internal jumps created by IF, FOR-NEXT, etc.

A statement such as:

```
IF A%>B% THEN A%=B%
```

Will compile into something like:

```
CMP A%, B%
BLE $1
MOV B%, A%
$1: …
```

Each branch (such as BLE in the above) is represented by three words in the intermediate code. One word encodes the type of branch needed, another holds a link to the previous branch, the third holds a link to the previous branch that references the same target. Thus, all the branches in the whole program are linked together in reverse order and all the targets are linked to the branches that can reach them.

The first step in branch optimisation is to look for conditional branches that jump around unconditional branches. These can be replaced by a single branch with the condition reversed.

For example:

```
BEQ $1     ; branch if equal
BR $2      ; branch (to somewhere else)
$1:
```

Can be replaced by:

```
BNE $2     ; branch if not equal
```

Also, any branch whose target is an unconditional branch to a second target can be replaced by a direct branch to the second target and any branch whose target is the next instruction can be removed altogether. These often occur because of the other optimisations.

PDP-11 conditional branches can jump a maximum distance of 127 words. If a longer jump is needed it's necessary to reverse the condition and branch around an unconditional long jump to the target (see example a little later).

BP3 optimises branches by traversing the (backward) list, computing the distance to the target of all forward branches. It determines those branches that can use the short form and those that may require the long form. This computation deducts the number of intervening NOP instructions encountered. During this backward scan, the links in the list are reversed so that the list can be rescanned in the forward direction to fix up the backward branches.

Finally, the code can be compacted, removing NOPs and fixing up the jump offsets for the branches. Any conditional jumps that are too far for the short form will be replaced by the long form. For example:

```
    BLT $1          ; branch if less than zero
```

Will be replaced by:

```
    BGE $2          ; branch if greater than or equal to zero
    JMP $1          ; long jump to the (distant) target
    $2: …
```

There is a further technique for shortening branches using a process called "branch-chaining" where it might be possible to use a short branch to another branch that has the same target. Although this can reduce code size, it generally results in greater execution time, so was not considered worth doing.

In the absence of floating-point hardware, all floating-point operations would have to be executed by calling run-time library subroutines. Since these have already been hand-crafted by DEC to be as efficient as possible, there is almost no scope for further improvement. However, the execution time of these subroutines would swamp any savings made by compiling the library calls.

Another problem with software floating-point is that most of the subroutines assume they can use the integer registers freely, so BP3s register optimisation would have been invalidated every time a floating-point instruction was encountered.

At the time BP3 was written, floating-point processors (FPP) were becoming increasingly accessible, with most of the recent PDP-11s having an FPP as standard, or available as an upgrade option.

It was therefore decided that BP3 would generate optimised code for the FPP instruction set, in a similar manner to integer code.

The FPP has six general registers but only the first four (F0-F3) are truly general-purpose. F4 and F5 are very tricky for a compiler to utilise, so BP3 preloads them with the constants 1.0 and 0.5. These are very handy for doing conversions, such as rounding. They also frequently appear as constants in user programs.

## Run Time System

BP3 has its own run time system. This is a complete re-write of the Basic-Plus libraries, taking account of the needs of BP3 programs and provides the extra capability of code-paging. The run time system occupies less than 8K words, allowing 24K words for the user program.

In the compiled code, each Basic-Plus statement has a line header. This contains the line number, since they are needed to allow ERL to be set correctly when errors are encountered. The header also has a pointer to the code. Since all PDP-11 machine instructions are word-aligned, there is a spare bit in every address. BP3 uses this bit is used to indicate whether the code is in memory or on disk. If the bit is clear, it's a pointer to the code in memory, otherwise it's the address of the code in the disk image.

The disk address can still only reference 32K but BP3 allocates the image file in 32-byte blocks and uses the pointer as a block number. This increases the addressable disk space to 1MB, at the expense of wasting a few bytes of disk space in the last block of each code section. The code header on disk contains the exact code size, so no memory is wasted when the code is loaded.

When the code space in memory is full, all the headers are reset to their disk addresses, effectively throwing away all the code that has been loaded. This may seem extravagant but it has been shown to be very effective. Most programs follow the 80-20 rule, 20% of the code is used 80% of the time. For example, a large part of many programs is the initialisation code that's executed only once.

Even after discarding code, the marginal time penalty for reloading each statement is quite small, so there is no great "hiccup" in execution time that can occur with garbage-collecting methods. Once a program settles into a major loop, very little reloading is needed.

Other paging techniques that rely on caching algorithms have their own drawbacks. They require extra overhead in both memory and time to determine which code should be discarded. Furthermore, selective discarding of code does not necessarily provide a large enough contiguous block of memory, so

some code-compaction, block-size sorting, or other garbage-collection is needed. Moving code during program execution can be problematic, requiring special techniques to adjust such things as return addresses held in the program stack.

The simplicity and effectiveness of throwaway caching greatly outweigh its drawbacks.

# Stuart McKenzie, Manx Computer Bureau

### RSTS and the Isle of Man

I first encountered RSTS in, I think, 1971 at a DEC training session in Reading. We bought our first PDP-11/40 in 1972 with RSTS-11 and we set up the first time-sharing computer bureau on the Isle of Man. Then we started the long task of creating bureau applications all written in Basic-Plus.

The company was called Manx Computer Bureau and we soon installed the first giant 40MB removable disks provided by the nascent hardware supplier Systime. It is amazing to see how far we have come hardware wise since those early days when you can now host 32GB on your watch.

RSTS/E was, in my opinion, miles ahead of its time and we had created real time accountancy systems in the early 70's. I say “real time” because most work then was on a batch basis but our leading package called MASTER allowed real time stock and accountancy control in that the entry of an invoice immediately affected stock levels and a trial balance was always available at any time of the financial cycle.

We programmed many packages over the years, particularly shipping applications such as maintenance scheduling or load planning for chemical carriers or voyage accounting but also stock and shares control, transport real time planning etc etc. The only real drawback was the speed of the proprietary hardware as we upgraded through the DEC releases up to twin 11/84s running 30 odd jobs dialled in by clients but this was the last use of PDPs and we shifted to large Alphas which allowed us to offer ISP facilities.

With the help of Datacrown we migrated the Basic-Plus programs to TBX under SCO Unix. When I retired in 2003, I brought an 11/84 home to use as a hobby machine. (Despite being “retired”, I still run the McKenzie Group of Companies in the Isle of Man.)

I still – yes STILL – run MASTER using an TBX emulator on Windows which must be one of the longest run Basic-Plus packages in existence and one which has definitely eliminated all bugs – and who can claim that these days?

Finally… a comparison: a month end run which took perhaps five minutes on an 11/84, admittedly with other programs running, now takes five seconds on an ASUS I5 and its the same number and complexity of reports. I also have a Threadripper 2990 32/64 beast on which I run Basic-Plus programs under TBX and boy is that quick but it is mainly used for ROSETTA Covid research!

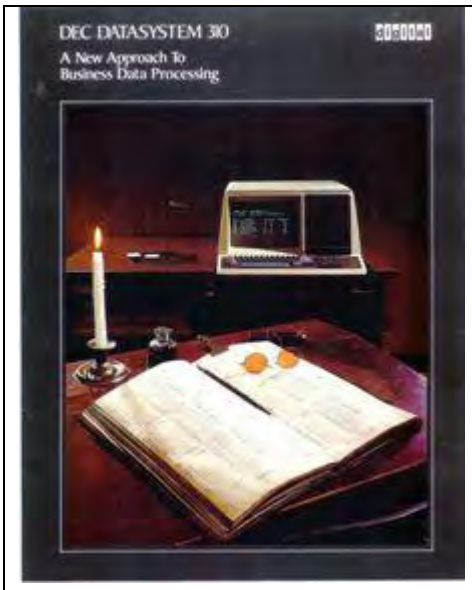
## Terry Dahler, Minicom, then freelance programmer

I started life as a COBOL programmer on Burroughs Mainframes.

In 1975 I applied for a job with Minicom Software Ltd of Blandford Dorset. During the short interview, I quickly discovered my future boss knew even less than I did. So I got the job … needless to say I was on a very poor salary!

At Minicom, my job was to develop commercial applications using DEC's Dibol software. Dorset was a particularly nice place to work but driving down in boss's TR6 was literally a white-knuckle ride: luckily we stayed in a pub so nerves could be calmed before work.

To eliminate any confusion, COBOL and Dibol had little in common except the last three letters of their name. See: I told you the boss knew nothing.




Our first customer was Raycroft in Finchley (spectacles / eyeglass suppliers), a joyous hour's trip (aka traffic jam) around the North Circular road, to develop an invoicing system.

The hardware was a DEC DATASYSTEM 310, a single seater/user machine (PDP 8, 16KB ram and 2 * 780KB RX01 8" floppy disks), mounted into a DEC supplied desk. Hot stuff. Note the synergy between DEC's brochure and Raycroft's spectacle business…

This limited configuration meant I had to overlay the program into four sections to fit into 16K. I can still remember the floppy disks chugging away as it swapped from customer account details to stock lines, to printing, the good old days... Programmers don't know their born nowadays... well that's my Victor Meldrew moment!!

Of course, the perks of a bit of additional tailored software resulted in several new pairs of glasses, with various tints. In those days' frames and prescription lens were less than £3… there goes Victor again.

By 1976 DEC allowed us to join the Big Boys School when they released CTS 300 which ran on a PDP 11 with Dibol under RT11 and with RK05's we had disks with a Megabyte number, heady stuff. This allowed multi-users and was very useful in developing lots of commercial applications for small businesses.

I can't quite remember when DEC allowed Dibol to be released under CTS 500 (RSTS), must have been early 80's. I think it is an example that while DEC always produced great technical equipment, maybe weren't so bright on the commercial front! Multiple commercial languages can hardly have been a bad thing, especially given the amount of software already under CTS300.

Most of my time on RSTS was spent moving Dibol software from CTS 300 to CTS 500. This gave us much larger disk capacity and a lot more reliability from the operating system. Getting many more users on the system also allowed us to sell to larger companies.

Harking back to the "good old days", I remember you could personalise your software then. Being a West Ham (Britain's greatest soccer club for American readers) supporter most of the error messages had more reference to the great team, along the lines of "Invalid Customer - West Ham for the Cup" or "Numeric only - Up the Hammers" and for customers on particularly friendly terms, "Numeric Only - Dumb Dumb", all squeezed into the last 40 chars on line 12/24 on the good old VT50/52/100/220 etc etc

Sadly when Xenix/Unix arrived, cheaper hardware made DEC kit too expensive and signed the beginning of the end for RSTS and VAX.

The last RSTS job I did was for Business Post, the courier company, to move the Dibol software to a Xenix 386 PC with disks four times the size and half the price, running 40 terminals. I think they refer to it as progress. It was then just a short step to world domination: Dibol programs running on a multi processor 486-based system supporting literally hundreds and then thousands of users.

## Thord Nilson Stacken Computer Club, Sweden

*Editor's note: The club, founded in 1976, is a part of the Student Union at the Royal Institute of Technology (KTH).*

RSTS/E First impressions

Before making contact with RSTS/E, I had done some programming in Uppsala-BASIC on an Alpha LSI and on a HP9830 BASIC calculator [1976-1977]. The BASIC on these machines where rather simple, variables could only be a single letter A-Z eventually followed by a single digit. IF statements did not have an ELSE clause etc. Apart from that some programming in machine code on an 8080 development board (MCS-80 KIT).

Then the company where I had a summer job in 1978 got a shiny new HP9845 - very expensive with graphics and a much more advanced BASIC. Long variable names, you could trap execution errors etc. Impressive!

Then the time had come for me to start study electrical engineering (EE) at The Royal Institute of Technology (R.I.T) in Stockholm, Sweden. This was September 1978.

In a small room in the basement were some teletypes model ASR33. They were connected to the EE educational computer, a PDP-11/45 with 248 KB memory. The default run-time system was Basic-Plus.

Being a bit sceptical I sat down at one of the free teletypes. Testing Basic-Plus I found it supported long variable names, IF statements with ELSE clause and error trapping with "ON ERROR GOTO" and much more. I was hooked.

Among other things, there was also an 8080 assembler on the system so I did no longer have to hand-assemble programs for my 8080 computer.

## The Stacken computer room B30.

The computer club had set up computer-room for a newly acquired KA-10 DECsystem-10 computer. This was in 1986. In this room there also was a PDP-15 and a PDP-11/40. The 11/40 had 248 KB of memory a disk controller with 2 * RP04 disks, a 1/2 in tape drive and a DECTAPE drive. We of course installed RSTS/E on it. It had some terminal lines hooked up to the institute's terminal exchange so you could connect to the machine from almost any terminal on the campus. A picture of the system can be found on page 13:

http://elvira.stacken.kth.se/~mz/sp/stackpointer-1986-5.pdf

First install was with the interpreted system cusps and the system ran reasonably well until some person decided that the precompiled cusps should be faster. However, as the new binaries were much larger and did not use the Basic-Plus Run Time System the machine started swapping instead so the machine ran much slower than before.

As a side note, the disk controller was built with flip-chips and took a whole cabinet. It also had a lot of indicator lamps blinking nicely when doing disk IO.

It was one problem though: it indicated a disk error. Being eager to start installing RSTS/E on the machine the flip-chip board that gated all the error sources together was pulled out and marked "to fix later". As the machine worked OK then I do not think “later” ever happened.

## The Stacken large computer room

At about the same time at my work we did a project, that ran from about March 1986 to April 1988, for one of the larger Swedish newspapers. The project was to replace the now outdated electronics for the distribution of the newspaper bundles from the presses to the distribution lorries. This old system ran on a PDP-8 with the programs overlayed on floppy disks. Also, the IO was centralized so literally a ton of cables entered the computer room. The new system should have distributed IO and was based on two Micro-VAX-II (for redundancy) that were connected to a network of nodes. The nodes were either nodes where bundles of newspaper were loaded on a train going around the room or nodes where the bundles where unloaded from the train to a specific distribution truck and then there were some service nodes. It was about 50 nodes totally. The nodes consisted of a 68008 processor with 256KB memory + MMU, a lot of I/O and a network interface.

To let the Micro-VAX communicate to a specific node an interface was needed. It was decided to implement this interface so it looked like a DZ11 interface to the VAX and then let it talk the network protocol to all the nodes. By doing this we did not have to develop any drivers for the board for VMS.

This also had the effect you could use any terminal comm program (for example Kermit) running on the VAX to communicate directly with the nodes for troubleshooting.

The Stacken large computer room was granted in 1988, then the work to make it ready took almost two years, so the opening was in 1990.

An extra DZ11 interface board (we called it MZV11) and a node computer came to good use when Stacken later moved to the large computer room. About 340m² with space for many computers, mostly DEC. The cooling of the room was dimensioned to be able to handle up to 400kW. For this there were three outside water to air coolers with four large fans each. There were also three circulation-pumps to circulate the cooling liquid.

As both the Micro-VAX and PDP-11/23 used Q-bus, the interface board could be used in the PDP-11.

The 68008 node was placed in the basement in an electrical cabinet with the contactors and inverters to control the fans and pumps. From the cabinet was

drawn about 60 meters network coaxial cable to the computer room where the PDP-11/23 with RSTS/E was located.

On the PDP-11/23 console you could see the status of the fans, pump and temperatures of the system and later also an estimation of the cooling power from the temperature difference and coolant flow.

A description of the system (in Swedish, from page 8) can be found here:

http://elvira.stacken.kth.se/~mz/sp/stackpointer-1990-3.pdf

A drawing of the computer room, see page 3:
http://elvira.stacken.kth.se/~mz/sp/stackpointer-1988-3.pdf

## Playing music with the PDP-11.

At the opening of the large Stacken computer room [1990] there should, of course, be some music. As a MIDI synth could be borrowed for the occasion there was a need for sending the midi-codes to the synth. The midi interface is actually a 5V, 5mA current loop running at a baud rate of 31250 baud. One of the machines that was up and running reliably at that time was a PDP-11/44 with RSTS/E V8.07. The serial interface on that had a maximum speed of 9600 baud with RS232 signal levels. So, a small interface was built with an 8048 microprocessor to convert levels and speed. Now the midi codes must be sent in real-time with millisecond precision for the music to be OK.

As RSTS/E is not a real-time OS the solution was to pad the midi-stream with no-ops consisting of two characters. As the time to send one character at 9600 baud is very close to 1 ms the time resolution would be about 2 ms. So a small conversion program to convert the ASCII midi file to binary with padding was written. The resulting binary file was then dumped to the midi-synth at 9600 baud via the 8048 interface. This worked really well.

## Trying RSTS/E on an old CAD-station.

We got an old CAD-workstation based on a PDP-11/73 with 256 KB dynamic memory. Installing RSTS/E on it went fine but it did not want to start the monitor. After some fiddling with the XBUF we finally got it to start.

One of the first things to test was to start PIP.

```
run $pip
?Odd address trap
?Program lost-Sorry
```

Oops, that not good but lets try the command again... same result. Tried a third time... same result. But why give up now? Tried it a fourth time and PIP started OK.

Examining the memory with SYSTAT showed three 1K locked out memory regions. After replacing three bad memory chips the machine ran OK. Do not remember when but probably around 1992-93.

## The wake-up call

I was working late one night. Time was somewhere between 1:00 and 2:00 in the morning when one of my colleges comes into my room. He is wondering if I can write a program to call him at home at a specified time to wake him up. Reason is that he often needs more than one call to wake up and that ordering that from the telephone company became expensive.

So, the wakeup program was created. Just submit it as at batch job set for the time you want to be woken up. It was created on 14-Jun-85 at about 2 in the morning.

## Some of our utility programs…

Yes, RSTS was more "roll your own utilities" and it was quite easy since you could do almost all from Basic-Plus. With VMS it was more just use the ready-made utilities.

That said, for our development machine at work, we did buy a lot of cross-assemblers and some c-compilers. One set of these were ported from Unix to RSX (mainly) so they ran under the RSX run time system. You could see that they came from Unix by the command line options. A lot of different single character options.

When entering a CCL command, the first thing RSTS does is to convert everything to upper case. Now Unix uses both upper and lower and lower case as the norm. So before starting the actual program everything is converted to lower case. It was also too many options for all to be lower case so some of them were in upper case. Thus one more option was needed: "treat the rest of the line as upper case", which converted the rest of the line back to upper case. As the options was far too numerous and hard to remember, I made the program ASM.BAS so you could write the options in a comment in your source code and ASM.BAS started the correct program with your specified options.

Elsewhere in this document, you will find a link to the simulator on Elvira. Some of the programs we wrote/used can be found on PUB:

```
8085  .BAS -- The optimized 8080 assembler, after a detour on VMS
ASM   .BAS -- Pre-processor for various (cross)assemblers
```

```
CPMLIB.BAS -- Manage 8in CP/M floppies
CLIB  .BAS -- Like CPMLIB but uses image files instead of floppy
CRUN  .BAS -- For starting DECUS-C programs
FORMAT.BAS -- Format floppy disks
LUNCH .BAS -- Send "time for lunch/coffee" messages
MINATO.BAS -- For communication with an E-prom programmer
TALK  .BAS -- Multi terminal chat using send/receive
TCOPY .BAS -- Program to copy tapes with only one tape drive
UNTAR .BAS -- For reading Unix tar tapes
VAKNA .BAS -- The "wake-up" program
VST   .BAS -- Set terminal modes for a vt100 compatible terminal
WHO   .BAS -- Who is online
```

## Modem Control

I was a student at R.I.T, so around 1980. I was discussing something with one of the sysadmins of Elvira when we were going home for the day quite early in the afternoon. While waiting at the bus stop, some other students came running saying that Elvira did not work and it needed to be fixed because there were classes later in the afternoon.

A new feature "modem control" had been enabled on all the terminals so that if you forgot to logout and then turned off the terminal the job got detached so that next user would not get your old session. Now there was a fast SYSTAT (FSY for short) that locked itself in the memory while displaying the status. So the other students had tested what would happen if you turn off the terminal in the middle of the printout and you repeat that procedure enough times? Result: memory full of locked jobs waiting for terminal output and no way to login or do anything from an already logged in terminal as all jobs were swapped out.

Needles to say the SYSTAT program was quickly patched.

## Bad memory day

One day I was going to do some fix-ups on one of my 8080 assembly programs. Entering the terminal room next to the computer room the normal hackers were also there. They say the machine had been acting strange lately. Anyway I sat down at a free LA36 DECwriter, starting to edit my code. When running it trough the assembler it complained about a few spelling errors. Fixed those and ran it trough the assembler again, this time with more errors. Not good. Programs started to crash with strange errors. Then one of the sysadmins comes and says we need to take down the machine.

At R.I.T there were many institutions that used PDP11s and as memory was very expensive at the time one of the institutes had designed its own add-on memory to the PDP11 and Elvira had some of that memory. Now this memory needed some different voltages that were not available in the back-plane so there was also some voltage converters needed.

So we all go into the computer room taking down the machine, opening the memory cabinet and examining the memory boards. Some of the capacitors for the voltage converters had started leaking.

So new CMOS memory had to be ordered. After this change the max uptime of the machine changed from 1-2 days to over a month, limited only due to the scheduled preventive maintenance. The machine also became some 10% faster. As RSTS was pretty good at restarting due to errors the low uptime had not been seen as a big problem.

## The last program

I was working in a small start-up, we were about 4-5 people doing consultant work on electronics, signal processing and programming. By 1984 we saw the need for a good computer to do the development work on. The choice came to be a PDP-11 with RSTS/E. One reason was that most of us were familiar with it from the institute (R.I.T.) were we were taking our masters degrees. A VAX computer would have been too expensive and too large for the office we had at the time and IBM-PC's were still so primitive.

By 1986 we had grown to 13 people and we saw the need for a program to keep track on all time spent on the various projects, generating invoices, keeping track of salaries for people only working part time etc.

So, I wrote a time-reporting program in Basic-Plus using the virtual arrays for databases. About 1 week programming effort for the first version and then about 1 week more for updates and maintenance over the years.

By 1995 we were no longer a separate company but a department of a daughter company to AtlasCopco and it was time to move to a new office. The time reporting should now be made in the new business system MOVEX. Intentia should just implement a time-reporting module with the current program as specification. So, the PDP-11 was left in the old place and the reporting was to be done via modem until the MOVEX module was ready.

Over one year later, by September 1996 I could run the final backup of the system and then shut it down. Before shutdown 29 people were using the system.

So Basic-Plus could be really efficient for development of small business programs.

# Tim Chamberlain, Datacrown UK

## The joys of being a RSTS software developer!

Having the (mis?)fortune of being able to write RSTS device drivers all too frequently ended up as very long working days...

Once Mike Mayfield had produced 'The LINK' (a remarkably clever piece of device driver software that allowed multiple slave RSTS systems to share the disks of a host RSTS system) in response the CPU bound problems of that era it was only a matter of time before a Systime user would want one.

Equally inevitably that user would be running Systime's home grown indexing package (ACCESS) which funnelled index lookup/update requests via RSTS's message send/receive to a detached program (ACCESD) that controlled the index file. For successful operation of ACCESS over The LINK it would be necessary to redirect those index requests from the remote systems across The LINK to the detached program on the host and then route the replies back to the requesting program on the correct remote host.

Clearly a new device driver that cuddled up to 'The LINK' device driver to provide message send/receive between RSTS systems was needed along with some low-level interactions with the host system's ACCESD from the remote systems. As the man with both RSTS internals and ACCESS guru status the pressgang from Silver Programs came knocking on my door.

A "few days" and nearly 5000 lines of code later the product was ready for the first ~~guinea-pig~~ esteemed paying customer, Astec based in the West Midlands who at that time were manufacturing power supplies (if my aged grey cells are to be trusted).

So, I popped down one weekend (30 years ago a motorway trip to Birmingham was queue free) to install and configure all the additional software. Initial testing went smoothly so it was back up the M6 for me...

...until the intermittent host system crashes started. Core dumps make boring reading especially when they show memory corruption with no clue as to how it became corrupted.

Darrin Haughton at Astec worked tirelessly to try and get the problem reproducible (thank goodness for nice customers). Eventually he could load the system enough for a failure to occur within a couple of hours. So that was me back on the M6 one evening for some night debugging.

About 4am the following morning we struck gold and had the problem narrowed down to the offending bit of code in 'The LINK' device driver that was stamping on a vital part of the data in the RSTS kernel reserved for my device driver.

The SYSGEN process for RSTS builds the driver data areas one after another in the order the drivers are defined and mine was straight after the LINK and so I was the unlucky one who got whacked over the head – nowadays we call these a "buffer overrun" probably because they're so common. Back then a "buffer overrun" probably meant the printer couldn't keep up and you lost bits of the text or got a hundred pages of hieroglyphics as the printer glitched into graphics mode.

To be fair back then such bugs were unusual as writing in machine code meant the catastrophic effect of such errors was normally immediate. A few months later Mike did mention that they'd had a problem with crashes when a host served two remotes and as a workaround they had to be configured the two devices as SS0: and SS2: because if they used SS1: that's when they got the system crashes. So, my device innocently sitting after SS0: was occupying the word of memory where SS1: would go and hence got clobbered – thanks Mike!

And for fun here is the patch that fixes the problem...

```
!                              LNKDVR V2.2
!
! Problem:  When allocating a UMR for a data transfer it is necessary to
!           supply a unique identifier tag. However HOST (server) systems
!           erroneously use the DDB address for the requesting DR unit
!           number -- this fails when there is more one DR unit
!
! Solution: The following patch corrects the problem by passing the MCB
!           address as the identifying tag. This value is unique for the
!           life of a transfer (i.e. as long as the UMR is required).
!
! Note:     This problem is ONLY apparent on UNIBUS systems with a 22-bit
!           address space (mainly the 11/44, 11/84, 11/70) which have been
!           configured as a HOST (server) for at least 2 REMOTES or if the
!           CSR/vector addresses for the HOST DR11 have NOT been entered
!           in the first slot (unlikely since the LNKGEN procedure uses the
!           first slot automatically).
!           The consequences of this problem are corruptions to disk data
!           either actually on disk (when writing) or in memory (when reading).
!           Such corruptions are not limited to user data (files and programs)
!           but may also affect system data (directories, loading overlays,
!           swapping etc.)
!
File to patch? <LF>
Module name? <LF>
Base address? SSDVR
Offset address? 6322
 Base   Offset  Old     New?
??????  006322  016304  ? NOP
??????  006324  ??????  ? NOP
??????  006326  004777  ? ^C
```

...I've always loved patches that remove code (NOP is machine code for 'No OPeration')

On the bright side Darrin was a perfect host, around 10pm he sent out for probably the best Dopiaza I've ever eaten – well it was Brum after all. AND in

thirty years he is the only customer who found the time to write rather than just say “thank you”.

ASTEC EUROPE LIMITED
High Street, Wollaston
Stourbridge, West Midlands DY8 4PG
Telephone Stourbridge (0384) 440044
Telex 337482 Cables: Electronic Stourbridge

Datacrown Limited
42 Halifax Road
Todmorden
Lancashire
OL14 5QG.

14th Feb 1990

Dear Tim,

Just a short line the thank you for your invaluable help over the past few weeks. I doubt that we would have a working three node Link operational if you had not appeared on the scene.

The door bell is working fine, but I have yet to try it in the Q-Bus PDP 11/83.

Once again, thankyou, and look forward to something nice in your Christmas stocking. (It's a lump of coal for NDS!)

Best regards to you and family.

Yours Sincerely,

Darrin.

Tim Chamberlain tim@datacrown.co.uk

## Tim Rhodes, Cumulus Systems, UK

Five snippets …

I contracted the illustrious Peter Dick (who I had met at a DEC conference) to expand a report generator Peter had written. I kept on asking Peter forever more challenging extensions. One day when examining a recently added piece of code, I came across the comment *'This line cost Tim a grand'*.

I subsequently acquired the product, for proceeds which enabled Peter to order yet another new Porsche.

Dateline 1973: Commodities processing software author Rolfe and Nolan (led by Malcolm Rolfe) took delivery of what DEC UK badged as the first 11/40 in the

UK for commercial time-sharing. A few months later Cumulus Systems (led by my good-self) installed delivery of (according to DEC) the first 11/45 in the UK for commercial time-sharing. DEC was already a significant player in what was called then 'time sharing' aka cloud / software as a service but it was a masterly exercise in milking deliveries for maximum PR benefit.

The small world of the PDP-11. In the early 70's I was involved with a conversion job and the code referred to software house which originated it. It turns out that the programmer was none other than Mr Porsche - who had been employed by Terry Williams. Terry subsequently became Chairman of Cumulus.

I contracted another well know PDP aficionado, Eoin Hanley, to write a system for Peter Hand: from the Feed Additives front end to the Least Cost Formulation.

Eoin's then business partner was Eric Bell who also did some work for me. I got an invoice for Eric's work from 'Control G Software Limited'. About a month later I twigged that the ASCII character produced by a control-G on a PDP console was the 'BEL' command, which rang a physical bell on some older terminals. Eric Bell … control-G geddit ??

Final memory: 25 years of Silver Christmas cards … followed by 18 years of Datacrown Christmas cards… all by the same cartoonist…

## Trevor White, Whitcom, UK

My first contact with RSTS/E was when introducing a new system into a builders merchant called Marlows in Bury St Edmunds where I worked as the computer manager – coincidentally Marlows has of yesterday announced it is closing the final part of its business although the PDP-11s and RSTE have long since gone – replaced by generic boxes hidden away in a cupboard somewhere!!

Marlows was one of the first adopters of a real time merchanting system from a company in Manchester called ACSL (Applied Computer Systems Limited) that wrote the original code and foolishly left the source code on the customer PDP-11s – me being fresh out of university and inquisitive (very early days) soon got to grips with Basic-Plus and discovered the delights of RSTS/E, even dabbling in Basic Plus 2 – must have been an easy to understand and use system as I actually did understand and use some of it – increasingly more as time went on.

This of course meant I would 'tinker' (or develop as I would call it) writing extra programs to do things that users thought would be a good idea – changing screen layouts, amending reports developing add on modules etc. As things developed more and more users needed access and so Marlows were early adopters of that fine piece of software and hardware – The Link. This really

opened things up – from memory (ha ha) we ended up with one PDP-11/70, two 11/84 and two 11/83s all linked together and over a hundred users.

Time for me to move on – so having set up the Link and ACSL software in Marlows there were other Builders Merchants out in the world that really needed my expertise (only they didn't know it as yet). So setting up my own company (Whitcom) and subcontracting work from ACSL whilst keeping Marlows running I set out into the world of self – employment (scary to start with as just married with a young son)- bought myself a 11/83 for the office and set about changing the world (or my bit of it anyway).

There were a lot of PDP-11s and RSTS systems out in the world and so went on my journey around the country helping builders merchants with their ACSL software, upgrading it and helping with installing various upgrades – the Link of course featured prominently as did other Silver Programs software and accessories – and so I met a certain Mr Dick and Mr Chamberlain – a course in somewhere up North followed and then an installation in Luton (builders merchant Gibbs and Dandy ) which kept me going for quite a few years developing software and systems for them.

With time and the development of bigger, faster and better PCs and windows and Unix based servers – the aging PDP-11s sadly had served their time so another course (Southend this time on Unix operating systems – think I still have the manuals somewhere).

BUT – hooray along came Trans_Basic to the rescue – we could all run our great software on Unix based computers faster and more efficiently – this opened up a whole new era and the ACSL system could live once again – and Whitcom could continue (especially as another child had arrived so we needed the cash !!).

Jacksons in Lincoln became great exponents of the Unix systems with Peter Clarke and Adrian Old leading the charge – me bringing up the rear (as usual) and plugging gaps in their resources when needed – I even managed to train Adam when he first came to Lincoln in the dark arts of Trans_Basic and Unix.

Of course at the head of it all was a certain Gentleman who I think I met first when helping with the Link installation and Marlows and later at Jacksons (in Lincoln) – of course trips in the Porsche could never be forgotten and our visit to Bally in the USA was again a highlight – how many casinos can one person visit in a night!! – all in the pursuit of software research of course – never forget that you should always play the outside of the row slot machines…

Again, time took its toll and a lot of my 'clients' and dare I say friends moved over to the dark side of Windows systems.

I have enjoyed my experiences with good old RSTS/E and its many spin offs but also with the many interesting characters I have met along the way and the friendships made.

RSTS – happy Birthday

## Walter Silvester, Contractor to Sterling Cranes
### Calgary, Alberta, Canada

Thinking of RSTS/E brings back memories of PDP-11/40, TU10, RP04 and so on.

It was technology that I think was ahead of its time.

Now 50 years later, our grandchildren's cell phones far surpass the capacity of a room full PDP, RSTS/E and Basic-Plus, etc. This operating system and hardware provided a good living for me for 25+ years. It was a pleasure work with state of the art systems at that time.

## ?Program lost-Sorry

*Editor's note: While tracking down historic users, I stumbled on a few aspects that caused my brain to fail with the classic error message.*

*https://rsts11.com/*

*Is the website of Robert Novak but has nothing to do with RSTS-11. He chose the name because he once worked on a PDP-11/44 for his programming/computer science coursework.*

*http://www.rsts.co.uk/*

*This website stands for "Richard Sierakowski Technical Services" and also has nothing to do with our RSTS.*

*http://www.rsts.com.sa/*

*RSTS is the leading Systems Integrator in Saudi Arabia. RSTS stands for Riyadh Saudi Technical Systems.*

*https://pdp-11.org.ru/files.pl?lang=en*

*Although this website appears to contain RSTS Version 9 distribution files (and the Y2K patches), I haven't checked any further as the ".ru" means the website is in Russia…*

*Anonymous Old Friend…*

*Editor's note: He was a System Analyst at DEC European Head Office in Geneva. This was the only DEC building in the world that allowed wine to be available in the staff canteen at lunch time. Happy times for contractors even if the staff ignored this freedom. Sorry. I digress. He was tasked with the specification of what became the internal OAS (Order Administration System) in the late 1970s that ran on RSTS/E systems.*

Unfortunately, I am not able to contribute to your notes on RSTS/E. To me, that was just another operating system; one of very many that my programmes ran on. I have no specific memories concerning RSTS/E.

*Hans Schilders*

*Editor's note: Hans was Dutch computer salesman in the 1970s and 1980s and currently he is both Chairman of the Dominica Hotel and Tourist Association and the Rotary Club. Hans replied to my email…*

Interesting email but no, I didn't work on a PDP-11. I worked in NCR and PDPs were part of our competition!!! Wishing you success with your collected memories.

*The winner of my personal challenge to find the earliest application system available with all the Basic-Plus source code…*

http://pdp-11.trailing-edge.com/rsts11/

Version September 1972
RSTS-11-46 North Country Union High School Administration Programs

This package contains both financial and student accounting systems. The financial system includes accounts payable, expense ledger, and payroll; the student administrative system includes attendance accounting, grade reporting and transcripts. The software runs on a PDP-11/20 with DECtape and RF/RS 11 disk, uses card input for all subsystems and an 80-column printer for output.

*Finally, thanks to Tim, if you are missing your RSTS system, here is one of the best simulators:*

http://elvira.stacken.kth.se/

```
Welcome to the PDP11 simulator.


RSTS V8.0-06 Elvira  Job 8  <Dial-up>  KB11  27-Jun-20  21:53

User: 99,99
```

```
Password:
Last logged in on 27-Jun-20, 21:50    at KB11:
Welcome to RSTS/E V8.0  Time Sharing at Elvira, Stacken
computer club.

This is a PDP11-emulator running RSTS/E V8.0 in commemoration
of the first computer "Elvira" that the students at KTH-E had
unlimited access to. The original Elvira was a PDP11/45 with
248 KB memory, running RSTS/E (mostly V6 and V7)

Some good programs to know about:

AMIS -- An early EMACS clone. Tutorial at [1,4]amis.tut
(terminal vt100 works on most terminal emulators, do "set speed
110" to get rid of annoying fillers.

KERMIT -- To transfer files.

Some other programs are located at PUB:
Do "run gam:advent" or "run gam:dungeo" for one of these
classical games.

Otherwise the system is quite vanilla.
Questions/comments, send mail to: thordn@stacken.kth.se

Ready

&right(sys(chr$(6)+chr$(9)+chr$(i%)),3%) for i%=0% to 127%
```

*Editor's note: Output not shown…*

```
bye/y
Saved all disk files on SY: 1280 blocks in use
Job 8 User 99,99 logged off KB11 at 27-Jun-20 22:02
System RSTS V8.0-06 Elvira
Run time was 2.6 seconds
Elapsed time was 9 minutes
Good evening
```

# RIP (RSTS In Peace)

Old Programmers Never Die, They Just Decompile

| | |
|---|---|
| Barry Isaacs | J.M.Hill and Sons |
| Bill Sconce | RSTS Development Team |
| Billy Maxman | Silver Programs Philadelphia |
| David Peel | British Steel / Quantic Computing |
| Dennis Mann | DEC Internal Systems |
| Chris Blake | Freelance Basic-Plus Programmer |
| Jack DelBalso | RSTS Development Team |
| John Sharp | Hill Samuel |
| Roger Box | Darkcrest / KDG |
| Simon S. Szeto | RSTS Development Team manager |
| Vernon Miller | RSTS Development Team |

Ends. 20-Jul-20