THE DNLS COMMAND LANGUAGE MONITOR.

This report is intended to describe the function and implementation of the DNLS command language monitor, to give examples of its use, and to propose a number of improvements and extensions for this type of analysis.

I. Command frequency monitoring in DNLS.

The DNLS system, as it runs today, is driven by a language comprising over 150 main commands - a "command" being understood here as either a one-letter or a two-letter code initiating a specific NLS operation. (note that some of these commands, for instance ej for Execute Journal or gq for Goto Query, may trigger entire subsystems that have their own command sub-structure. The sub-structures are beyond the scope of this analysis.)

For the purposes of display, these commands can be roughly divided into "editing commands" and "subsystems", although these names are not perfectly accurate.

In the first group we find a matrix of 8 operations (copy, delete, move,...) that apply to 11 qualifiers: branch, character, group, etc. All combinations in this matrix are valid commands, including such obscure processes as "Xset Invisible". The table of editing commands therefore contains 88 elements. In addition we include in the first table such one-letter commands as Append, Break, Null file, Quit and Viewspec.

Table 1.

	brn	chr	grp	inv	lnk	num	plx	sta	tex	vis	wrd	tot
copy	-	-	-	-	-	-	-	-	-	-	-	-
delt	-	3	-	-	-	-	-	-	-	-	-	3
inst	-	2	-	-	-	-	-	2	1	-	1	6
move	-	-	-	-	-	-	-	-	-	-	-	-
repl	-	1	-	-	-	1	-	-	1	-	1	4
subs	-	-	-	-	-	-	-	-	-	-	-	-
trsp	-	-	-	-	-	-	-	-	-	-	-	-
xset	-	-	-	-	-	-	-	-	-	-	-	-

appd 0
brek 1
null 1
quit 0
vspc 1 pt= 45300 rt= 782422

The second group is also a matrix of seven operations that may apply to 19 qualifiers, but not all entries are valid commands. The operations are: Execute, Freeze, Goto, Jump, Load, Output and Update.

Table 2.

	Exec	Frez	Goto	Jump	Load	Outp	Updt
a	-	-	-	-	-	-	-
b	-	-	-	-	-	-	-
c	-	-	-	-	-	-	-
d	-	-	-	-	-	-	-
e	-	-	-	-	-	-	-
f	-	-	-	-	1	-	-
h	-	-	-	-	-	-	-
i	-	-	-	6	-	-	-
j	-	-	-	-	-	-	-
l	-	-	-	-	-	-	-
m	-	-	-	-	-	-	-
n	-	-	-	-	-	-	1
o	-	-	-	1	-	-	-
p	-	-	-	-	-	-	-
q	-	-	-	-	-	-	-
r	-	-	-	-	-	-	-
s	-	-	-	1	-	-	-
t	-	-	-	-	-	-	-
u	-	-	1	-	-	-	-

In the implementation of an internal counting mechanism for these commands it is not feasible to restrict the analysis to the combinations shown above in Tables 1 and 2, because NLS is an evolving system where new commands appear and disappear; since the beginning of this phase several commands have been added and two have been taken out (Execute WSI measurements and Execute 940 file). The analysis package uses a 30x30 matrix to accumulate command counts, so that it becomes a simple matter to reflect such changes.

In the following we describe how the package can be used as a statistical tool by the individual user (section II) through a set of simple commands, and how we intend to put it to work in a general analysis of DNLS usage (section III). In the last part (section IV) recommendations are offered for the design of the next phase.

II. Using the command analysis package on-line.

In the early design stage for this work there was some discussion of the existing statistical commands in NLS. The discussion is documented in the Journal (12793,) (13110,) (13143,) (13191,) and led to the decision to separate timing studies of the Superwatch type from command usage analysis. The "Execute WSI Measurements" command was taken out.

The analysis package which is available for on-line use is part of

the "Goto Use Measurements" sub-system and is called by typing the two letters: gu.

There were two sub-commands at this point (Begin and End Measurements) that triggered a timing mechanism which is probably obsolete. (Question: is this code really obsolete and, if so, could we speed up NLS by taking the measurement flag test out of all core-NLS routines?). Pending some decision on this code, I have not touched the "Begin" and "End" subcommands. Remember, however, that they are NOT part of the command usage mechanism, which is ON automatically and requires no user action to initialize it.

G[oto] U[se Measurements] F[requency count]

is the command that gives a user access to the current state of the matrix. This may be followed by one of the commands:

D[isplay]
T[ables]
R[eset]
S[ave

The Display command is intended for debugging and maintenance purposes rather than for general use. It shows the state of the entire string of counters, in rows of ten numbers with a running index to the left

The Table option overlays the current status of Table 1 over the user's display (which is preserved and can be restored with a Command Delete). Hitting a CA (command Accept) will call Table 2

The Reset command will reinitialize the counting machinery

The Save command will i) automatically create a file named QBVMXYZ.LAN in the user's directory if none exists, XYZ being the user's ident, ii) write Table 1 and Table 2 as NLS statements in that file in LIFO fashion, iii) update the file and iv) reset all counters.

The file that results from use of the Save option is under user control and can be edited, formatted and processed like any other NLS file.

This set of commands therefore provides a flexible mechanism for monitoring one's own use of DNLS, running special statistical experiments, etc.

III. Automatic statistics-gathering.

Whenever a DNLS session terminates normally, the system executes code that has an effect similar to that of the Save command described above. The statistics that have been accumulated during the session are written out as NLS statements in that user's analysis file.

A user program named "Sweeper" that we intend to run every night will gather up these statements in a single master file and reset the user files so that directory space problems are minimized.
The information gathered into the master file is the following: For each DNLS session,

1) The user ident
2) The date and termination time of the session
3) rt, the real time duration of the session
4) pt, the CPU time used
5) The count statistics obtained for all commands.

All times are expressed in milliseconds.

IV. Proposal for next phase.

The question arises of processing the information in the master file in order to produce meaningful statistics on the utilization of DNLS commands. The following is a proposal along these lines.

1. Daily,

A printout of the master file will be produced with viewspecs y and K (statement signatures).

A consolidated table of command usage could be generated.
Time distributions reflecting session duration (both CPU and real time) could be printed out.

2. Weekly,

A table showing the distribution of most common DNLS commands for the ten largest users would seem to be a useful result. It would give some indication of variability in user behavior (programmers vs. non-programmers, for instance).

A general table of commands with their overall usage frequency would also be useful, together with a separate table of commands that have not been used at all during that period.

A general user population profile would be a good way of graphically reflecting command usage.

Individual profiles could also be generated, using as a basis the

command ranking derived from observation of the entire user population. (See example below from my own experiments).

3. Monthly,

We could again produce consolidated tables and profiles. Some tables could be selected for their usefulness in timing studies. For example, assuming we had ten tables giving CPU time and real time on the basis of ten frequently-used commands, we could solve the linear system to obtain the time coefficients for each command, thus producing a very accurate measure of command efficiency.

4. Example:

This proposal can be illustrated by the consolidated tables and a profile obtained from the three DNLS sessions that were required to enter and edit Parts I through IV of the present report.

Table 3.

	brn	chr	grp	inv	lnk	num	plx	sta	tex	vis	wrd
copy	-	-	-	-	-	-	-	-	-	-	-
delt	-	7	-	-	-	-	-	-	-	-	1
inst	-	10	-	-	-	-	-	30	1	-	4
move	-	-	-	-	-	-	-	1	-	-	-
repl	-	5	-	-	-	1	-	-	3	-	4
subs	-	-	-	-	-	-	-	-	-	-	-
trsp	-	1	-	-	-	-	-	-	-	-	-
xset	-	-	-	-	-	-	-	-	-	-	-

appd 2
brek 1
null 0
quit 0
vspc 4 pt= 307 sec. rt= 5789 sec.

The table above was obtained by adding together the elements of three matrices resulting from the three separate DNLS sessions. Similarly we can derive a table of usage for the main subsystems:

Table 4.

	Exec	Frez	Goto	Jump	Load	Outp	Updt
a	-	-	-	-	-	-	-
b	-	-	-	-	-	-	-
c	-	-	-	-	-	-	-
d	-	-	-	-	-	-	-
e	-	-	-	2	-	-	-
f	-	-	-	-	3	-	-
h	-	-	-	-	-	-	-
i	-	-	-	11	-	-	-
j	-	-	-	-	-	-	-
l	-	-	-	-	-	-	-
m	-	-	-	-	-	-	-
n	-	-	-	-	-	-	4
o	-	-	-	4	-	-	-
p	-	-	-	-	-	-	-
q	3	-	-	-	-	3	-
r	-	-	-	-	-	-	-
s	-	-	-	7	-	-	-
t	-	-	-	-	-	-	-
u	-	-	3	-	-	-	-

These tables lead to the following ranking of commands:

Table 5.

30	is	5	rc	4	iw	3	eq	1	rn
11	ji	4	v	3	rt	2	a	1	it
10	ic	4	rw	3	lf	2	je	1	ct
7	dc	4	jo	3	oq	1	dw	1	ms
7	js	4	un	3	gu	1	b		

And they result in the following profile:

```
is  ********************************
ji  ***********
ic  **********
dc  *******
js  *******
rc  *****
v   ****
rw  ****
jo  ****
un  ****
iw  ****
rt  ***
lf  ***
oq  ***
gu  ***
eq  ***
a   **
je  **
dw  *
b   *
rn  *
it  *
ct  *
ms  *
```

These measurements indicate that in the course of the three sessions, 24 commands appeared, of which 18 were used more than once, and six more than five times. The timing studies also lead to an estimate of the overall cost of producing such a document, which (at current BBN rates) would be of about $50 for computer cost alone.

There are probably many other useful results that could be derived from the information contained in the master file as described above. In a future phase, the timing studies could be refined and they could be combined with an analysis of command sequences.

V. Results of monitoring the experimental system.

During the first weeks of January, the monitoring facility was implemented in the experimental system. We saved the measurements obtained in a limited number of sessions by five different users. Their primary activity was text-editing and viewing. Although the total amount of connect time was small (about eight hours) the results show some interesting facts. The following table lists the commands that account for 50% and 75 % of overall usage.

Table 6: Experimental system.

	JFV	CHI	MDK	JFV	-PR	JFV	total	CHI	JBN	DVN	total	cumul
ji	11	6	5	4	18	3	47	1	4	29	81	81
is	30		14			8	52	3	2	8	65	146
ic	10		3		8		21	3		22	46	192
jl		5		1		1	7	3		27	37	229
v	4			7		1	12	6	12		32	261
js	7	3		2		5	17	2	1	7	27	288
lf	3		1	6	1	5	16		4	7	27	315
dc	7		8		5		20			3	23	338
jo	4			3	2	3	12		1	7	20	379
rc	5		2		2		9	1	1	6	17	396
it	1		1				3			15	17	413
rw	4	1	3				8	1		7	16	429
jb										16	16	445
un	4	2	1	3		1	11	2	1	1	15	460
rt	3				1		4		1	7	12	472
a	2		1		8		11				11	483
gu	3			1			4			6	10	493
jc				2	1		3			7	10	503
jr		2					2			7	10	513

In the next table are the commands that were used less than ten times in this series of sessions, i.e. those commands used in the lower 25% of the command usage.

Table 7. Experimental system (cont.) Lower 25%

iw	4	1	3				8			1	9
ds					3		3	1	1	4	9
oq	3			1	1	3	8			1	9
dt					4		4			4	8
ju		2	1	1	1	3	6			1	7
gp								3		4	7
eq	3		1	1			5		1	1	7
mb			1		2		3		1	3	7
mg					6		6				6
je	2			1	1		4			2	6
jn				4			4			2	6
gd				5			5				5
mc			1		1		2			3	5
jd						1	1			4	5
ms	1	1	1				3	2			5
n				4			4				4
od						1	1			3	4
dg		1	1				2	2			4
jp					2	1	3		1		4
b	1	1	1		1		4			1	4
rn	1						1			3	4
q		1				2	3				3
ej								1		2	3
es										3	3
ct	1						1			2	3
jt										3	3
dp				3			3				3
ja				2			2				2
cs		1					1		1		2
cb				2			2				2
el					1		1		1		2
tc					1		1			1	2
cg					1		1				1
mt										1	1
cc						1	1				1
dw	1						1				1
ed		1					1				1
uo										1	1
jh					1		1				1
xw										1	1
fr									1		1
rv								1			1
tn										1	1
to	115	29	49	53	72	41	359	36	34	252	681
pt	307	79	89	126	101	89	791	55	70	442	1358

rt	5789	977	1621	1281	2082	4574	16324	1444	2011	8382	28161
r1	2.7	2.7	1.8	2.4	1.4	2.1	2.2	1.5	2.0	1.7	1.9
r2	50.3	33.4	33.1	24.1	28.8	111.5	45.4	40.1	59.1	33.2	41.3

r1: process time per command, seconds.
r2: real time per command, seconds.

From this experiment we derive a first evaluation of the average cpu time per DNLS commandd (1.9 seconds) and of the average real time per command: 40 seconds. It is interesting to compare these figures to similar measurements under normal running conditions.

VI. Results of monitoring in the running system.

The measurement system was turned on in the running DNLS and statistics were gathered for eleven users during an entire day (January 22). The master file was then analyzed leading to the statistics of Tables 8 and 9 below. (Sessions by Kirk were distributed into two groups, KK1 and KK2.)

Table 8 shows the commands representing 50% and 75% of total usage.

Table 8. Running System.

DNLS command usage statistics for one day.

user	JEW	DVN	JCN	JBN	EKM	MDK	KK1	KK2	CHI	JAK	MFA	DCE	total
1 ji	21	5	59	21		3	122	39	1	17	7	8	303
2 is	11		6	19		11	58	33	1	2	2	1	144
3 rw	2		29	2			44	29			2	2	110
4 lf	12	4	25	23		3	5	6		3		12	93
5 ic	7	3	7	6			31	31	2	3	1	1	92
6 jl	17	9	19	7	3		6	3	6	1	5	3	79
7 jf	3	5	18	6			16	12			7	6	73
8 iw	2		1				42	18		2			65
9 ms	4			1		1	28	26					60
10 gd	5	2	9	2			23	12		2		1	56
11 rc	4		3	9			13	16		6	2	2	55
12 dc	2		7	13		3	12	11	1	3	2		54
13 js	11	1	12	5		11	3	3	3				49
14 jo	5	1	18	12			7	3		2			48
15 ds	4	2		4		1	10	9	1	16			47
16 gu		1	30				13						44
17 jr	3	4	8				20	5				1	41
18 jp	2			5	1		21	6	1	3		1	40
19 rt	1		1	8		1	6	12		10			39
20 dw	3		1				27	6					37
21 dt	3		1	14			9	6				2	35
22 mb	3	1	2	9		12	8						35
23 v		1	11	21			1						34
24 it				23		8				3			34

Table 9 lists the commands accounting for the remaining 25%:

Table 9. Running system (cont.)
DNLS command usage statistics for one day.

user		JEW	DVN	JCN	JBN	EKM	MDK	KK1	KK2	CHI	JAK	MFA	DCE	total
25	uo		2	11				12	8					33
26	ju	2		1			1	20	5				2	31
27	b	5		3	2			9	8					27
28	jh			2			2	15	7			1		27
29	a	1	3	2				5	9			1		21
30	mg	3		1				8	8					20
31	cs	1		1	1			11	6					20
32	oq	2	2	5	2		2	1	1	2			1	18
33	gp	6		2		3				4			2	17
34	eq	1	3	1	6	4					1			16
35	ct			10			2	1	1					14
36	cw	1						7	6					14
37	el	1		1	1		1	2	3	1	1	1	2	14
38	xw			2				7	5					14
36	db			2	5				2			3	1	13
37	un	6			5		1		1		3	1	6	11
38	of	1		1	2			3	3		1			11
39	eu			10									1	11
40	je	2	1		1	2		3	1					10
41	ja	2						8						10
42	rs							3	6					9
43	dg	1	1				1	2	3					8
44	mc			1	3			2	2					8
45	mt	1					1	3	2					7
46	mp	2						3	2					7
47	jc	5						1	1					7
48	gs			4			3							7
49	dv	1						2	3					6
50	rv	3						1	2					6
51	cc				1			3	2					6
52	jd		3		1				1		1			6
53	xc			1	1			4						6
54	jt		1					2	1		1		1	6
55	jb	1	1					2	1					5
56	rg							4	1					5
57	cj		1					1	2	1				5
58	od			2	3									5
59	n	2		1	1									4
60	ed	3							1					4
61	mw							3	1					4
62	xs				1			3						4
63	tw							3					1	4

64	q		2						2				4
65	fs	2		2									4
66	cb			2			2						4
67	tc	1	1					1					3
68	jn				1		1		1				3
69	er		3										3
70	cg	1					1	1					3
71	di					2		1					3
72	sp	2						1					3
73	tt						1	2					3
74	cv			2									2
75	cp							2					2
76	gn	1							1				2
77	dl	1					1						2
78	rp						2						2
79	iv						1	1					2
80	tv			2									2
81	fr				1								1
82	fa			1									1
83	ip		1										1
84	ii		1										1
85	ea			1									1
86	ei										1		1
87	sv			1									1
88	cl						1						1
89	tb						1						1
90	ts							1					1
91	mi							1					1
92	ri							1					1
93	rl	1											1
94	rb									1			1
95	sw							1					1
96	gb								1				1
total		187	65	342	246	13	68	690	404	29	82	36	57	2219
pt		419	386	475	418	88	60	778	587	102	300	65	96	3774
rt		6833	1373	4812	16243	3550	618	23240	24105	720	4787	1061	2294	
r1		2.2	5.9	1.4	1.7	6.8	0.9	1.1	1.4	3.5	3.7	1.8	1.7	1.7
r2		36.5	20.5	13.6	65.5	273.1	9.1	33.1	58.8	58.8	58.4	29.5	40.2	40.4

total cpu time: 1 hour.
total connect time: 25 hours

average cpu time per DNLS command: 1.7 seconds.
average real time per DNLS command: 40 seconds.

CONCLUSIONS

Monitoring of the DNLS command language both in experimental and in normal running conditions shows that less than twelve commands account for 50% of overall usage, and about 23 commands account for 75% of total usage.

Table 10. Usage of subsystems.

	Exec	Frez	Goto	Jump	Load	Outp	Updt
a	1	1		10		-	
b	-		1	5			
c	-		-	7		-	
d	4		56	6		5	
e	-		-	10			
f	-			73	93	11	
h				27			
i	1			303			
j	5						
l	14			79	-		
m	-		-				
n	-		2	3			11
o	-			48			33
p			17	40			
q	16					18	
r	3	1		41			
s	-	4	7	49		-	
t	-			6			
u	11		44	31			
TOT	55	6	127	738	93	34	87

Table 11. Usage of Editing commands.

	brn	chr	grp	inv	lnk	num	plx	sta	tex	vis	wrd	total
copy	4	6	3	-	1	-	2	20	14	2	14	66
delt	13	54	8	3	2	-	-	47	35	6	37	205
inst	-	92	-	1	-	-	-	144	34	2	65	338
move	35	8	20	1	-	-	7	60	7	-	4	142
repl	1	55	5	1	1	-	2	9	39	6	110	229
subs	-	-	-	-	-	-	3	-	-	1	1	5
trsp	1	3	-	-	-	-	-	1	3	2	4	14
xset	-	6	-	-	-	-	-	4	-	-	14	24
TOT	54	224	36	6	4	-	14	285	132	19	249	1023

appd 21
brek 27
null 4
quit 4
vspc 34

Total cpu time: 1 hour
Total connect time: 25 hours

It is interesting to observe how close the estimates of cpu time per command and connect time per command are found for the experiemntal and the running conditions, respectively 1.9 vs. 1.7 and 41.3 vs.40.4 seconds.

We propose to periodically conduct such analyses of command frequency (every six months, for instance) in order to monitor changing usage patterns of our system.

One possibly fruitful application of this set of statistics would be to speed-up the DNLS processor. The results presented above (especially those of Tables 10 and 11) indicate that there is a set of about twelve to twenty commands that could be regarded as primary candidates for such an optimization effort.

Figure 1

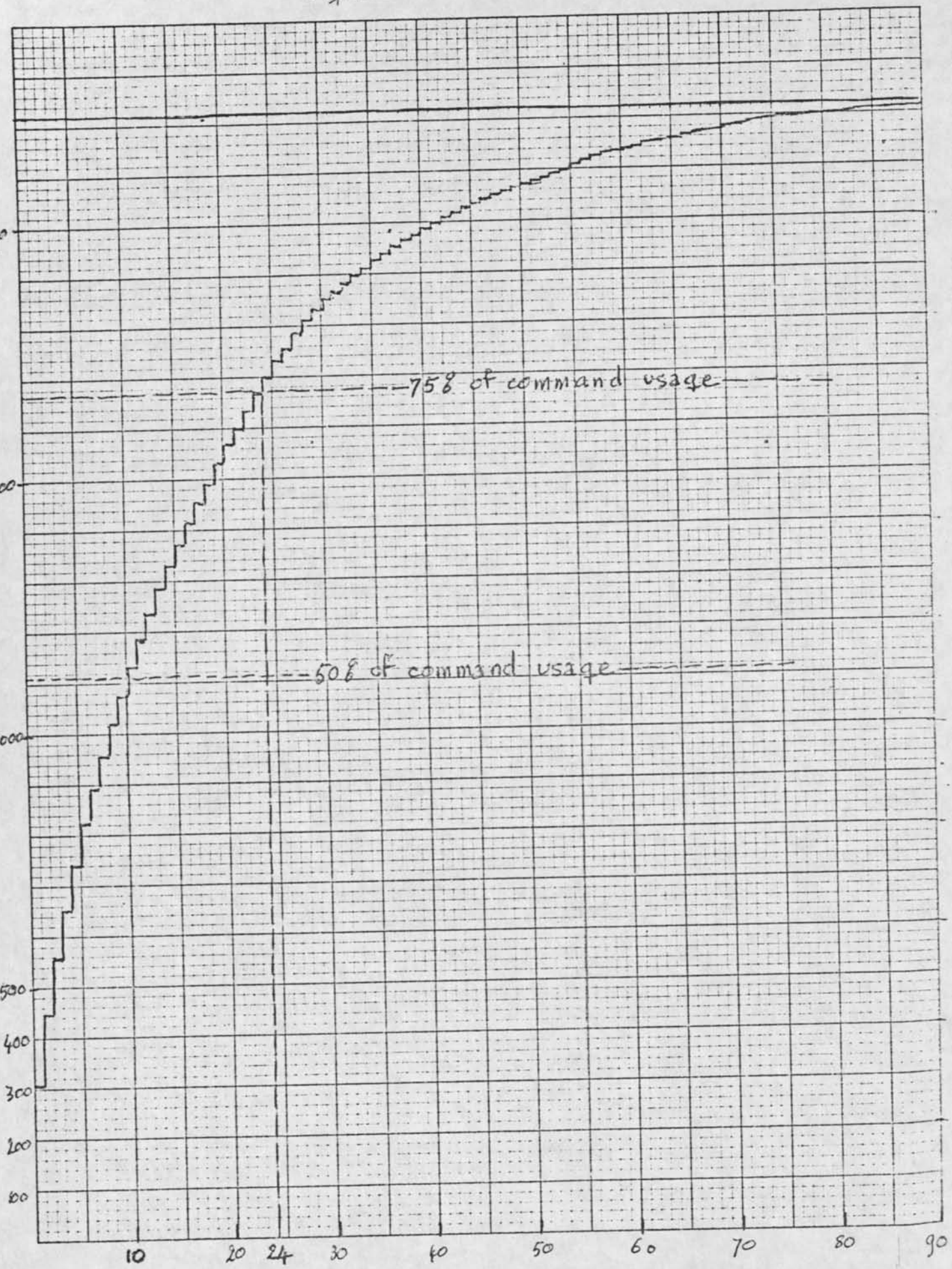
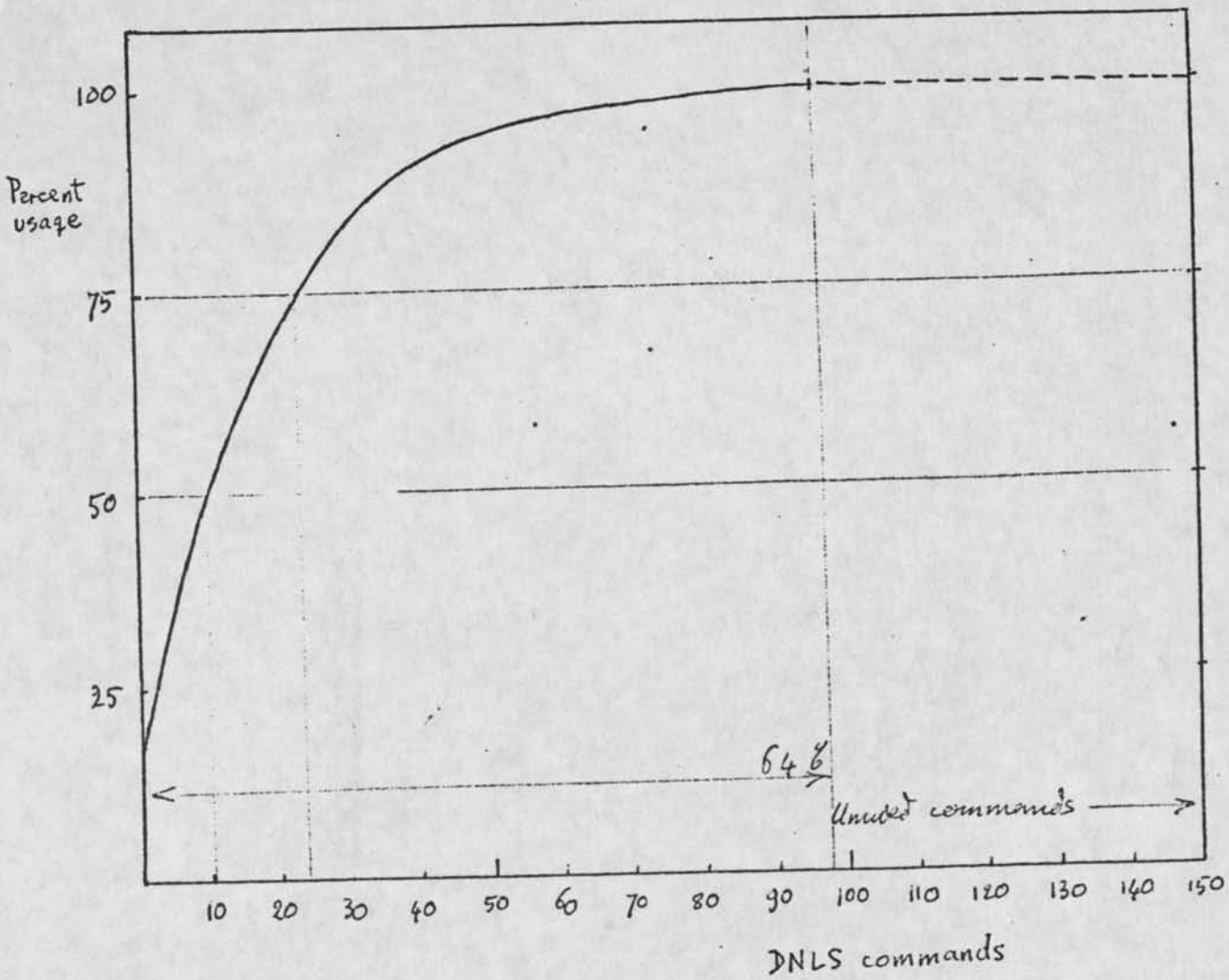


Figure 2


100
75
50
25
Percent usage
64
Unused commands
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
DNLS commands