Using Unix: Collected traces of 168 users

Saul Greenberg

Advanced Technologies
The Alberta Research Council
6815 - 8 St NE
Calgary, Alberta, Canada T2E 7H7

phone: (403) 297-2674 / email: saul@noah.arc.cdn

Abstract

The task of collecting long-term data on real-life computer use is onerous. There is the technical difficulty of writing monitoring software, the social inertia of attracting individuals to the study, and the often overwhelming politics of obtaining an organization's permission for executing the study. To save other researchers the time and effort required to collect such data, this paper describes how command line data from 168 users of the UNIX[1] *csh* system was collected and organized. The accompanying magnetic tape includes all the data files.

1 Introduction

This report describes how command line data was collected from 168 users of the UNIX *csh*. All data is included in the enclosed magnetic tape. The tape is made publically available for several reasons. It will save other researchers the time and effort required to collect long-term data. It will allow my own work to be replicated by others. It will allow us to compare and contrast analyses and results produced by different researchers.

The data was originally collected as part of a PhD project (Greenberg, 1988). Amongst other things, the resulting dissertation includes a comparison of collection methodologies used by other researchers studying UNIX, as well as a survey of their work. Excerpts from the dissertation are found in two papers. The first describes how people use commands in UNIX, while the second reports on the statistics of the complete command line entered by the user, rather than just the command itself (Greenberg and Witten, 1988a and 1988b). In the later, particular attention is paid to how users repeat their command lines during their system interactions.

The next section reviews briefly why studies of UNIX are worthwhile. The subsequent section describes how data was collected for the current study, while the final section describes how the data on the accompanying tape is formatted.

[1] Unix is a trademark of A T & T Bell Laboratories.

2 Why study Unix?

Observations of everyday human-computer interaction can, at least in principle, give valuable insight into people's behaviour when using computers. One popular vehicle for such studies is the UNIX operating system.

Studying UNIX is attractive for a variety of reasons. First, it is not a contrived "toy" system. Rather, it is widely used, very powerful and potentially complex, and has a broad range of users (Kraut, Hanson, and Farber, 1983). Because it is a general-purpose computing environment fulfilling many needs, any results garnered from it may generalize to other systems. In contrast, many high-performance graphical interfaces are so customized to particular applications that generalizations would be difficult to make and support.

Second, if UNIX findings could not be generalized, they would still be valuable in their own right. Although old, UNIX is far from dying. Rather, it is being rapidly disseminated as a *de facto* open system standard on diverse machines, running the gamut from mainframes to workstations and personal computers. Even users of graphical direct-manipulation interfaces thirst for UNIX, as illustrated by Apple's Macintosh/UNIX fusion. Vendors are now trying to modernize UNIX by embedding it within a window environment. The Sun workstation, for example, has a suite of window-based front ends to popular UNIX facilities, including the shell, debugger, mail system, terminal emulator, and so on (Sun Microsystems, 1986).

Another appeal of UNIX to researchers is that it has already been studied extensively. There is probably more knowledge and raw data on UNIX usage than any other computer system. The scientific process is more easily realized; other UNIX studies can be replicated; and previous findings can be built upon.

Finally, a pragmatic advantage of studying UNIX is that it is relatively easy to do, since large groups of diverse people use it at many different sites. Although generally perceived to be expert-oriented, there is no question that a significant number of non-programmers with widely varying needs also harness its powers. UNIX is often the standard system employed by research institutions. The benevolent setting allows large-scale realistic studies that span user categories. At the University of Calgary, for example, UNIX is used heavily in the Department of Computer Science by people with quite diverse programming skills and personal requirements. It is also available to people in several non-computer departments. The academic setting not only provides a captive audience, but also encourages participation — bureaucratic procedures are in place for conscripting subjects for study.

3 Data collection for the current study

In this study, command-line data was collected from users of the UNIX *csh* command interpreter (Joy, 1980). The selection and grouping of subjects, and the method of data collection, are described below.

Subjects. The subjects were 168 unpaid volunteers. All were either students or employees of the University of Calgary.

Name	Sample size	Total number of command lines	Number of command lines excluding errors		
			total	mean	std dev
Novice Programmers	55	77423	73288	1333	819.8
Experienced Programmers	36	74906	70234	1950	1276.0
Computer Scientists	52	125691	119557	2299	2022.9
Non-Programmers	25	25608	24657	986	1155.6
Total	168	303628	287736	1712	1498.8

Table 1: Sample group sizes and statistics of the command lines recorded

Subject use. Four target groups were identified, representing a total of 168 male and female users with a wide cross-section of computer experience and needs. Salient features of each group are described below, while the sample sizes (the number of people observed) are indicated in Table 1.

Novice Programmers. Conscripted from an introductory Pascal course, these had little or no previous exposure to programming, operating systems, or UNIX-like command-based interfaces. Such subjects spent most of their computer time learning how to program and use the basic system facilities.

Experienced Programmers. Members were senior Computer Science undergraduates, expected to have a fair knowledge of programming languages and the UNIX environment. As well as coding, word processing, and employing more advanced UNIX facilities to fulfill course requirements, such subjects also used the system for social and exploratory purposes.

Computer Scientists. This group, comprised of faculty, graduates and researchers from the Department of Computer Science, had varying experience with UNIX, although all were experts with computers in general. Tasks performed were less predictable and more varied than other groups, spanning advanced program development, research investigations, social communication, maintaining databases, word-processing, satisfying personal requirements, and so on.

Non-programmers. Word-processing and document preparation was the dominant activity of this group, made up of office staff and members of the Faculty of Environmental Design. Little program development occurred — tasks were usually performed with existing application packages. Knowledge of UNIX was the minimum necessary to get the job done.

Since users were assigned to subject groups only through their membership in identifiable user groups (*eg* Computer Science graduate students), their placement in the categories above cannot be considered strictly rigorous. Although it is assumed that they generally follow their group stereotype, uniform behaviour is not expected.

Instructions to subjects. As part of the solicitation process, subjects were informed verbally or by letter that:

- data on their normal UNIX use would be monitored and collected at the command line level only;
- the data collected would be kept confidential;
- any public reference or dissemination of the data and derived results would guarantee anonymity, unless explicit permission was given by the subject to do otherwise;
- at any time during the study period the subject could request that data collection stop immediately;
- there would be no noticeable degrading of system performance;
- if requested, data collected from a subject would be made available to him or her.

Subjects did not require nor did they receive any additional instructions during the actual study period. No subject asked to be withdrawn from the experiment, and no-one asked to see their personal data.

Apparatus. A modified *csh* was installed on three VAX 11/780's located in the Department of Computer Science and one VAX 11/750 in the Faculty of Environmental Design, both within the University of Calgary. Many different terminals were available to participants, most which were traditional character-based VDU's. In addition, Corvus Concept workstations running the Jade Window Manager were available to members of the Experienced and Computer Scientist groups (Greenberg, Peterson and Witten, 1986). This workstation allowed users to create many "virtual terminal" windows, each running *csh*, on a single screen.

Method. Command-line data was collected continuously for the four months between February 1987 through June 1987 from users of a modified Berkeley 4.2 UNIX *csh* command interpreter (Joy, 1980). From the user's point of view, monitoring was unobtrusive — the modified command interpreter was identical in all visible respects to the standard version. The total number of command lines recorded per group are listed in Table 1.

Data was collected by recording lines expanded by *csh*. Instead of collecting data by catching keystrokes as they are entered, the complete line submitted was captured as a chunk after it has been entered and processed by *csh*. Extra information known to *csh* was trapped and recorded as well by placing "hooks" within *csh*. History use and alias use are noted, as well as the current working directory of the user and the error status after execution is attempted. Table 2 lists the trace information annotated by the modified *csh*. Login sessions are distinguished by a record that notes the start and end time of each session (the 'S' and 'E' fields in the Table). Command lines entered during this period are then listed in following records, each annotated with the current working directory, alias substitution (if any), history use and error status. The final command line accepted by *csh*, including history expansions and ignoring editing operations that form the line, is recorded in the 'C' field. The 'D' field notes the directory the user was in when the command line was entered. The alias expansion of the line is found in the 'A' field, while the 'H' field indicates whether or not *csh* history helped form the line. System errors generated by *csh* are registered in the 'X' field. These annotate eleven categories and many sub-categories of errors. The code definitions are provided in Table 3. The total and average number of command lines collected excluding these

Code	Description	Example
	Login session record	
S	Start time of the login session	S Fri Feb 6 15:54:25 1987
E	End time of the login session	E Fri Feb 6 17:25:01 1987
	Command line record	
C	The line entered by the user	C ls –a
D	The current working directory	D /user/greenberg/bin
A	The alias expansion of the previous command (if any)	A ls –a
H	The line entered had a history expansion in it (T for true or NIL for false)	H T
X	The error detected in the line by *csh* (if any). A following letter and number code indicates the category and actual error type.	X N 10

Table 2: Trace information annotated by the modified *csh*

errors are listed in Table 1. Table 4 illustrates an example fragment from an imaginary trace file.

The Appendix at the end of this paper provides summary statistics for each subject, which includes the number of login sessions, the command lines entered, the different commands used, the *csh* errors noted, the times history was used, and the different directories accessed.

Data Selection. If subjects did not log in at least ten times and execute at least 100 commands during the study period, their data was not considered. By these criteria, 12 of the 180 original participants were rejected.

Motivation. Participants used UNIX as usual. Users were neither encouraged nor expected to alter their everyday use of the system. As subjects had few reminders that their command-line interactions were being traced, they were largely oblivious to the monitoring process.

Confidentiality of data. All subjects were promised confidentiality and anonymous references. To this end, each trace files was modified by replacing the subject's names with x's. For example, if a command line in smith's original trace file was "cd smith", it was changed to "cd xxxxx".

Problems and limitations. Tracing lines expanded by *csh* is a tradeoff between recording too much and too little information. Several problems and limitations of the data collection method employed here are noted below.

First, due to implementation difficulties, the details of history directives were not recorded. The altered *csh* indicates only that history has been used, and notes the command line retrieved through history. It does not record the actual history directive used to produce the modification.

Main error categories							
A	alias problem	D	directory error	J	job error	R	redirection problem
B	built in problem	E	expression error	M	reg expression error	S	syntax error
C	control error	H	history problem	N	execution error	Y	system error
Error sub-categories, as written to the screen by Csh							
0	unknown error	31	Ambiguous output redirect	62	Improper mask	93	Interrupted system call
1	*chdir didn't work	32	Can't ≪ within ()'s	63	No such limit	94	I/O error
2	No other directory	33	Ambiguous input redirect	64	Improper or unknown scale factor	95	No such device or address
3	Directory stack not that deep	34	Badly placed ()'s	65	Bad scaling; did you mean ?	96	Arguments too long
4	Bad directory	35	Invalid null command	66	Can't suspend a login shell (yet)	97	Exec format error
5	Directory stack empty	36	Ambiguous	67	Can't from terminal	98	Bad file number
6	No home directory	37	$< line too long	68	Not login shell	99	No children
7	Can't change to home directory	38	No file for $0	69	Unknown user:	100	No more processes
8	Usage: dirs [-l]	39	Subscript out of range	70	Path error	101	Not enough core
9	No match	40	Bad : mod in $	71	Missing]	102	Permission denied
10	Command not found	41	≪ terminator not found	72	Arguments too long	103	Error 14
11	Unmatched (something)	42	Line overflow	73	Pathname too long	104	Block device required
12	Word too long	43	Divide by 0	74	Unmatched '	105	Mount device busy
13	Variable syntax	44	Mod by 0	75	Too many words from	106	File exists
14	Expansion buf ovflo	45	Expression syntax	76	Undefined variable	107	Cross-device link
15	Bad ! form	46	Missing }	77	Usage: jobs [-l]	108	No such device
16	No prev sub	47	Missing file name	78	Bad signal number	109	Not a directory
17	Bad substitute	48	Too few arguments	79	Unknown signal; kill -l lists signals	110	Is a directory
18	No prev lhs	49	Too many arguments	80	Arguments should be jobs or process id's	111	Invalid argument
19	Rhs too long	50	Too dangerous to alias that	81	There are stopped jobs	112	File table overflow
20	Bad ! modifier	51	Empty if	82	No current job	113	Too many open files
21	Modifier failed	52	Improper then	83	No previous job	114	Not a typewriter
22	Subst buf ovflo	53	Syntax error	84	No such job	115	Text file busy
23	Bad ! arg selector	54	Not in while/foreach	85	No job matches pattern	116	File too large
24	No prev search	55	Invalid variable	86	No job control in this shell	117	No space left on device
25	: Event not found	56	Words not ()'d	87	No job control in subshells	118	Illegal seek
26	Alias loop	57	then/endif not found	88	No such file or directory	119	Read-only file system
27	Too many)'s	58	endif not found	89	Error 0	120	Too many links
28	Too many ('s	59	endsw not found	90	Not super-user	121	Broken Pipe
29	Badly placed (	60	end not found	91	No such file or directory	122	Disk quota exceeded
30	Missing name for redirect	61	label not found	92	No such process		

Table 3: Error codes annotated by the Unix *csh*

Line in trace file	Comment
S Fri Feb 20 23:39:46 1987	*Session starting time*
E Fri Feb 20 23:59:31 1987	*Session end time*
C who	*The line entered to csh*
D /user/cpsc500/xxxxxx	*The current directory*
A who | more	*"who" is an alias for "who | more"*
H NIL	*History was not used*
X NIL	*The line did not generate a csh error*
C cd ~cpsc500	*The line entered to csh*
D /user/cpsc500/xxxxxx	*The current directory*
A cd ~cpsc500 ; set prompt="$cwd:t]!> "	*"cd" is an alias*
H NIL	*History was not used*
X D 69	*A csh error was produced, classified as a directory error (code D). More specifically, an unknown user (code 69) was given in the directory path.*
C who	*This line was recalled via history (see H)*
D /user/cpsc500/xxxxxx	*the current directory*
A who | more	*"who" is an alias for "who | more"*
H T	*History used*
X NIL	*The line did not generate a csh error*
S Tue Feb 24 23:41:39 1987	*A new login session*
E NIL	*And so on ...*

Table 4: A fragment from an imaginary trace file

unix-data/			
README			
unix-data/experienced-programmers/			
experienced-1 *through* experienced-36			
unix-data/novice-programmers/			
novice-1 *through* novice-52			
unix-data/non-programmers/			
non-1 *through* non-24			
unix-data/computer-scientists/			
scientist-1 *through* scientist-52			
unix-data/show-error-code/			
Makefile	error-code.c	error-code.h	error-code

Table 5: Directory structure of the data files on tape

Second, and more seriously, not all user activity is captured. Although recording *csh* lines works well for "batch" style programs that execute and return without user intervention, it does not capture activity within the interactive applications used (*eg* editors). Interactive information is lost since data is collected from the *csh* command line only. Also, commands cannot be considered "equal." For example, consider a trace containing only two UNIX commands: *ls* for listing files; and *emacs* which invokes a sophisticated interactive editor. Whereas file listing is accomplished almost immediately, an editing session can last for hours. This distinction is not captured here.

Third, the actual processes spawned by the command line are not noted. There are many ways to execute programs in UNIX; directly by name, indirectly through an alias or *csh* variable, or as a suite of programs through a script. Because of this diversity, users can invoke the same program by many different names. For example, *e, emacs* and *ed* may all invoke the same editor. As only the text typed to *csh* is collected, the actual processes executed is left as an educated guess.

Implementation note. Since the source for *csh* contains over 16000 lines of sparsely documented and quite complex code, the task of modifying it was quite difficult. Four months were required to produce an acceptable tested version of *csh* that included a robust monitoring facility, even though the final number of modifications required was relatively small. This time includes the bureaucratic red tape involved with obtaining *csh* source.

4 The Data Tape

The enclosed tape was made on a SUN using the UNIX command "tar cbf 126 /dev/rst8". Its contents can be extracted in the usual way. For example, "tar xf /dev/rst8" will extract all the data files, while "tar tf /dev/rst8" will list the tape's contents. The tape has also been read on an Apollo running Unix, using the command "tar xf /dev/rct8".

The tape contains approximately 21 megabytes of data, arranged as files in a hierarchical directory as described in Table 5. The parent directory is called unix-data. The README file contains a brief summary of the tape contents — a hardcopy is included with every tape. The subdirectories experienced-programmers, novice-programmers, non-programmers and computer-scientists contain all the data files, one for each subject. The directory show-error-code contains a C program, called error-code, that converts the letter and number error codes in the X field with their textual equivalent. This program reads the data file from standard input and writes to standard output.

Acknowledgements

Ian Witten provided valuable advice, help, and encouragement throughout this project. This report was completed during a National Science and Engineering Research Council postdoctorate appointment at the Alberta Research Council.

References

Greenberg, S. (1988). *Tool use, reuse, and organization in command-driven interfaces.* PhD thesis, Department of Computer Science, University of Calgary, Alberta. (Research Report No. 88/336/48)

Greenberg, S. and Witten, I.H. (1988a). Directing the user interface: how people use command-based systems. In *Proceedings of the 3rd IFAC Conference on Man-Machine Systems*, Oulu, Finland, June 14-16.

Greenberg, S. and Witten, I.H. (1988b). How users repeat their actions on computers: principles for design of history mechanisms. In *Proceedings of the ACM SIGCHI '88 Human Factors in Computing Systems*, pages 171–178, Washington, D.C., May 15-19.

Greenberg, S., Peterson, M. and Witten, I.H. (1986). Issues and experiences in the design of a window management system. In *Proceedings of the Canadian Information Processing Society Edmonton Conference*, pages 33–50, Edmonton, Alberta, October 21-23.

Kraut, R.E., Hanson, S.J. and Farber J.M. (1983). Command use and interface design. In *Proceedings of the ACM SIGCHI '83 Human Factors in Computing Systems*, pages 120–124, Boston, December 12-15.

Joy, W. (1980). *An introduction to the C shell, volume 2c. Unix Programmer's Manual*, University of California, Berkely, California, seventh edition.

Sun Microsystems (1986). *Windows and window based tools: beginner's guide.* Sun Microsystems, Inc., Mountain View, California.

Main error categories							
A	alias problem	D	directory error	J	job error	R	redirection problem
B	built in problem	E	expression error	M	reg expression error	S	syntax error
C	control error	H	history problem	N	execution error	Y	system error
Error sub-categories, as written to the screen by Csh							
0	unknown error	31	Ambiguous output redirect	62	Improper mask	93	Interrupted system call
1	*chdir didn't work	32	Can't ≪ within ()'s	63	No such limit	94	I/O error
2	No other directory	33	Ambiguous input redirect	64	Improper or unknown scale factor	95	No such device or address
3	Directory stack not that deep	34	Badly placed ()'s	65	Bad scaling; did you mean ?	96	Arguments too long
4	Bad directory	35	Invalid null command	66	Can't suspend a login shell (yet)	97	Exec format error
5	Directory stack empty	36	Ambiguous	67	Can't from terminal	98	Bad file number
6	No home directory	37	$< line too long	68	Not login shell	99	No children
7	Can't change to home directory	38	No file for $0	69	Unknown user:	100	No more processes
8	Usage: dirs [-l]	39	Subscript out of range	70	Path error	101	Not enough core
9	No match	40	Bad : mod in $	71	Missing]	102	Permission denied
10	Command not found	41	≪ terminator not found	72	Arguments too long	103	Error 14
11	Unmatched (something)	42	Line overflow	73	Pathname too long	104	Block device required
12	Word too long	43	Divide by 0	74	Unmatched ‘	105	Mount device busy
13	Variable syntax	44	Mod by 0	75	Too many words from	106	File exists
14	Expansion buf ovflo	45	Expression syntax	76	Undefined variable	107	Cross-device link
15	Bad ! form	46	Missing }	77	Usage: jobs [-l]	108	No such device
16	No prev sub	47	Missing file name	78	Bad signal number	109	Not a directory
17	Bad substitute	48	Too few arguments	79	Unknown signal; kill -l lists signals	110	Is a directory
18	No prev lhs	49	Too many arguments	80	Arguments should be jobs or process id's	111	Invalid argument
19	Rhs too long	50	Too dangerous to alias that	81	There are stopped jobs	112	File table overflow
20	Bad ! modifier	51	Empty if	82	No current job	113	Too many open files
21	Modifier failed	52	Improper then	83	No previous job	114	Not a typewriter
22	Subst buf ovflo	53	Syntax error	84	No such job	115	Text file busy
23	Bad ! arg selector	54	Not in while/foreach	85	No job matches pattern	116	File too large
24	No prev search	55	Invalid variable	86	No job control in this shell	117	No space left on device
25	: Event not found	56	Words not ()'d	87	No job control in subshells	118	Illegal seek
26	Alias loop	57	then/endif not found	88	No such file or directory	119	Read-only file system
27	Too many)'s	58	endif not found	89	Error 0	120	Too many links
28	Too many ('s	59	endsw not found	90	Not super-user	121	Broken Pipe
29	Badly placed (	60	end not found	91	No such file or directory	122	Disk quota exceeded
30	Missing name for redirect	61	label not found	92	No such process		

Table 3: Error codes annotated by the Unix *csh*

Appendix: Summary statistics for each subject

Novice subject number	Login sessions	Total command lines	Different commands	Errors noted by *csh*	Times history was used	Different directories used
novice-1	55	2457	67	213	37	18
novice-2	118	1267	22	58	0	11
novice-3	345	2337	26	93	0	1
novice-4	61	1919	32	123	0	4
novice-5	62	593	24	67	0	5
novice-6	74	871	23	44	0	1
novice-7	94	1039	38	51	98	11
novice-8	92	1822	13	19	0	3
novice-9	44	853	26	63	0	6
novice-10	64	1464	42	40	0	3
novice-11	59	256	26	21	2	1
novice-12	438	2436	19	210	0	2
novice-13	49	652	20	49	0	2
novice-14	156	3194	67	208	0	27
novice-15	79	1139	14	48	0	1
novice-16	16	256	12	25	0	1
novice-17	135	1194	23	59	0	1
novice-18	46	1088	15	38	0	1
novice-19	103	3401	59	363	7	4
novice-20	54	418	18	19	1	2
novice-21	44	849	22	42	48	3
novice-22	122	1893	43	51	0	3
novice-23	90	2138	30	72	0	2
novice-24	86	849	26	53	0	3
novice-25	169	2066	13	217	0	1
novice-26	87	1120	19	60	0	1
novice-27	71	1195	25	63	1	9
novice-28	123	2221	31	120	0	1
novice-29	94	1230	14	44	0	3
novice-30	78	946	20	28	0	3
novice-31	64	2073	27	102	0	7
novice-32	51	385	20	37	0	3
novice-33	199	3127	31	106	0	6
novice-34	123	1276	25	46	4	1
novice-35	90	1444	22	54	0	6
novice-36	141	3213	55	137	0	5
novice-37	88	1949	36	57	0	32
novice-38	109	839	12	17	0	2
novice-39	74	1107	34	51	0	3
novice-40	58	967	17	24	0	5

Novice subject number	Login sessions	Total command lines	Different commands	Errors noted by *csh*	Times history was used	Different directories used
novice-41	86	2317	15	51	0	1
novice-42	92	1068	31	33	0	3
novice-43	33	608	18	26	0	1
novice-44	59	1277	14	40	0	2
novice-45	54	651	17	16	0	1
novice-46	276	4163	120	372	112	58
novice-47	56	1316	19	78	0	3
novice-48	23	269	12	9	0	1
novice-49	23	723	20	31	0	1
novice-50	48	985	33	92	0	3
novice-51	42	480	20	20	0	2
novice-52	69	650	22	38	0	3
novice-53	98	1028	34	41	0	1
novice-54	38	683	19	56	0	10
novice-55	62	1662	25	40	6	2
experienced-1	137	3714	74	298	174	58
experienced-2	25	219	28	11	6	8
experienced-3	28	915	51	42	88	16
experienced-4	151	3776	59	123	2	29
experienced-5	283	4015	78	222	35	44
experienced-6	53	757	56	32	0	17
experienced-7	189	5857	139	612	67	100
experienced-8	134	2930	74	265	67	54
experienced-9	99	2351	99	136	86	25
experienced-10	25	446	45	26	1	18
experienced-11	98	1456	43	86	21	48
experienced-12	66	1763	70	92	28	17
experienced-13	49	1109	60	160	25	30
experienced-14	103	1810	60	153	23	27
experienced-15	14	225	21	12	0	32
experienced-16	41	795	33	22	24	22
experienced-17	85	2343	67	144	0	32
experienced-18	25	575	27	21	5	9
experienced-19	122	1807	84	88	163	20
experienced-20	180	4556	79	370	435	44
experienced-21	100	2394	76	83	157	54
experienced-22	149	2814	67	122	325	18
experienced-23	95	2306	70	119	189	18
experienced-24	114	3331	132	228	222	62
experienced-25	71	1465	63	89	11	19
experienced-26	30	679	33	66	0	22
experienced-27	219	1693	70	54	77	43
experienced-28	440	3893	93	60	78	24
experienced-29	71	2214	59	133	59	67
experienced-30	130	2028	64	110	82	18

Novice subject number	Login sessions	Total command lines	Different commands	Errors noted by *csh*	Times history was used	Different directories used
experienced-31	68	683	82	38	19	40
experienced-32	65	974	72	87	47	32
experienced-33	59	1292	55	65	83	14
experienced-34	116	1869	59	218	206	15
experienced-35	165	4272	77	169	28	40
experienced-36	60	1580	70	116	56	54
scientist-1	165	1856	105	111	54	43
scientist-2	198	2954	87	149	236	37
scientist-3	133	978	38	69	1	6
scientist-4	238	4507	112	320	178	114
scientist-5	197	1563	77	78	18	13
scientist-6	145	1103	61	49	33	46
scientist-7	13	366	49	28	0	25
scientist-8	61	842	39	51	0	5
scientist-9	256	4067	89	65	224	42
scientist-10	129	2024	63	120	77	96
scientist-11	38	205	24	13	0	1
scientist-12	105	2499	117	52	53	63
scientist-13	108	3593	45	118	357	25
scientist-14	202	3433	109	183	23	83
scientist-15	161	1429	94	81	200	30
scientist-16	74	326	31	29	0	5
scientist-17	95	569	33	38	0	1
scientist-18	144	2831	71	112	106	74
scientist-19	189	5584	65	240	6	62
scientist-20	225	2697	112	189	74	52
scientist-21	81	1762	82	134	50	102
scientist-22	132	750	45	39	0	12
scientist-23	324	3360	91	135	52	48
scientist-24	72	1494	41	55	0	5
scientist-25	415	3508	112	122	7	113
scientist-26	123	983	65	70	0	24
scientist-27	111	3817	97	85	102	79
scientist-28	111	765	64	26	20	17
scientist-29	134	2683	60	243	20	61
scientist-30	180	2129	77	123	186	56
scientist-31	65	250	20	20	9	3
scientist-32	78	601	36	20	0	9
scientist-33	24	325	16	12	0	3
scientist-34	204	2639	61	88	15	50
scientist-35	80	1049	46	29	23	22

Novice subject number	Login sessions	Total command lines	Different commands	Errors noted by *csh*	Times history was used	Different directories used
scientist-36	275	12056	181	566	488	202
scientist-37	121	4187	61	83	121	64
scientist-38	131	3775	92	168	48	113
scientist-39	119	1753	76	77	173	40
scientist-40	348	4605	66	98	0	42
scientist-41	204	2037	49	36	0	5
scientist-42	298	6068	133	644	6	158
scientist-43	108	3106	86	101	0	37
scientist-44	72	1543	62	84	12	16
scientist-45	40	862	76	59	17	17
scientist-46	294	2551	92	110	80	89
scientist-47	75	1229	67	81	9	61
scientist-48	76	819	27	43	0	2
scientist-49	105	1448	108	97	138	46
scientist-50	138	1496	75	225	219	18
scientist-51	74	910	43	67	0	51
scientist-52	263	7705	121	299	231	93
non-progs-1	95	1622	61	59	0	7
non-progs-2	53	454	16	15	0	2
non-progs-3	85	1265	38	15	9	7
non-progs-4	133	5050	70	161	18	89
non-progs-5	77	244	8	11	0	1
non-progs-6	23	177	17	7	0	2
non-progs-7	80	1231	53	54	3	9
non-progs-8	23	239	32	13	28	14
non-progs-9	73	357	34	23	4	3
non-progs-10	32	495	36	20	0	21
non-progs-11	281	1848	27	61	0	17
non-progs-12	24	216	19	26	0	4
non-progs-13	30	487	10	5	0	1
non-progs-14	17	201	9	4	1	3
non-progs-15	78	571	15	28	0	2
non-progs-16	46	821	32	26	18	11
non-progs-17	61	848	19	65	0	1
non-progs-18	97	1403	22	64	0	2
non-progs-19	77	175	15	7	0	2
non-progs-20	137	4042	81	124	165	30
non-progs-21	25	132	5	7	0	1
non-progs-22	151	1567	39	56	48	8
non-progs-23	89	1294	47	48	0	5
non-progs-24	35	542	25	34	0	1
non-progs-25	76	327	9	18	3	1