LABORATORY FOR COMPUTER SCIENCE
(formerly Project MAC)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TM-89

MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM

HUMBERTO RODRIGUEZ, JR.

AUGUST 1977

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

MIT/LCS/TM-89

MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM

Humberto Rodriguez, Jr.

May 1977

MIT/LCS/TM-89

MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM

by

Humberto Rodriguez, Jr.

May 1977

The research reported here was sponsored in part by Honeywell Information Systems Inc., and in part by the Air Force Information Systems Technology Applications Office (ISTAO), and by the Advanced Research Projects Agency (ARPA) of the Department of Defense under ARPA order No. 2641 which was monitored by ISTAO under contract No. F19628-74-C-0193.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LABORATORY FOR COMPUTER SCIENCE
(formerly Project MAC)

CAMBRIDGE MASSACHUSETTS 02139

MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM

by

Humberto Rodriguez, Jr.

Submitted to the Department of Electrical Engineering and Computer Science on May 12, 1977, in partial fulfillment of the requirements for the Degree of Bachelor of Science.

ABSTRACT

One of the problems in measuring the performance of a computer system is in defining its normal workload. In the case of timesharing systems, it is necessary to develop a behavioral model of the average user. This thesis presents a study of several parameter that characterize user behavior on the Multics timesharing system at MIT. Data was gathered by monitoring the logon sessions of three different groups of users. The results are presented and comparisons are made between the command usage of the groups. Some patterns of usage do appear in the results, but it is unclear if they can be applied in other situations.

A probability distribution of the think time between commands is shown and compared with other distributions. The benchmark program currently used on the Multics system is also compared with the user model described in this study. The capability to monitor user behavior and characteristics is shown to be useful and worth installing in the system.

THESIS SUPERVISOR: Liba Svobodova
TITLE: Assistant Professor of Electrical Engineering and Computer Science

ACKNOWLEDGEMENT

I would like to thank my thesis supervisor, Professor Liba Svobodova, for her guidance and suggestions in getting this project started and in the presentation of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I - INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The area of research in computer performance measurement and evaluation is important to understanding the behavior of complex systems. Several techniques have been developed for measuring computer performance, but most of them have in common the problem of defining a system's normal workload. Since most systems, especially timesharing ones, interact with human users, it becomes necessary to define the behavior and characteristics of the user community in order to formulate such a workload. For example, in analytical or simulation models of timesharing systems a key parameter is the time a user spends betweeen requests, also known as "think time". In the case of benchmarking, where a simulated workload is put on an existing system, even more information is necessary about user characteristics.

There have been several studies that have tried to measure the behavior and characteristics of users of a timeshared system, most notably the one done by Boies (1974). This study was done on an IBM Time Sharing System (TSS/360) at an IBM Reasearch Center and measured a wide variety of user characteristics. One of the major problems with these studies is whether the results are generalizable, that is, do the users of System A behave the same as the users of System B? What if the systems differ greatly in ease of use, number of commands, command language flexibility, etc. ? The results may even vary between different installations of the same system. Another factor which may need to be measured is whether the work done by one sub-group of users is different from that done by another sub-group. For instance, how significant is the difference in work done by experienced programmers and beginners? Such a study was recommended by Treau (1974), but one of the difficulties is in deciding a priori which sub-group a user should be classified in.

This thesis is an attempt to measure several parameters which characterize the typical Multics user. The primary motivation is that such a study has not been done on the Multics system, although there was a study done at MIT on the CTSS system by Scherr (1965). Some of the benefits of doing this research are:

1). Currently, new versions of the Multics system are compared with older versions by running a benchmark. This presently consists of running several absentee users working on five slightly different scripts. The commands in the scripts mostly invoke different language translators and the think times between commands are selected randomly. It is questionable whether these scripts are a reasonable model of user behavior. Roach (1974) presented a critical review of the scripts, however, his observations focused on the problem that the benchmark does not load the system sufficiently. By obtaining command usage counts and think time distributions, a set of scripts can be designed that is more representative of normal usage and thus may be able to duplicate a normal workload.

2). By obtaining frequency counts of the commands, it would be possible to determine the set of most heavily used commands and decide how much effort should be spent in optimizing these commands.

3). With enough data on the needs and behavior of the users, a reasonable user model can be formulated which would be a great aid in any modeling and simulation studies of the system. For instance, in his PhD. thesis, Sekino (1972) had to determine the user think time. This parameter was important to his analytical model of Multics and without it, he was unable to examine the details of the validity of his model. Since he had no tool that would allow him to measure the think time of Multics users, he had to estimate this parameter by "general observation."

II - PROCEDURE

2.1 DIFFICULTIES IN MONITORING USERS ON MULTICS

There are two basic approaches that can be taken in monitoring what transactions a user performs at a terminal. One way is to intercept every line that is typed by the user, affix the current time to it, and store the data on some mass storage device. With this method, it is possible to study users' behavior in great detail. There is very little overhead for processing and some systems have such a trace facility built into them. The major drawback to this method is that so much data is gathered that it becomes necessary to dedicate some secondary storage device, such as a tape drive, for the duration of the monitoring. The second approach is to decide before the experiment which parameters are of interest and to specifically measure these as they occur. The data can be statistically accumulated during the monitoring so that the data storage area need not be very large. The major limitation to this approach is that once the study has been performed, there is no way of measuring any parameter that was not included, thus having to replicate the study. Also, some time dependent parameters would be difficult to measure such as repetitive groups of commands. Another disadvantage is the increased overhead of monitoring since some processing must be done to collect and identify the data for each command.

For the purposes of this study, it was decided that the second method would be the most feasible alternative on the

Multics system, despite its disadvantages. For instance, since Multics is primarily an on-line time-sharing system, there are not too many available tape drives and the cost of dedicating one for the duration of monitoring would be prohibitive. Also, despite the fact that Multics has an excellent set of built-in metering tools, there are no tracing facilities that allow the system to access all input lines from terminals. Another important limitation to this study was that the system software could not be modified in any way. This restriction meant that not all users could be monitored, only those who agreed before-hand to participate. Despite these drawbacks, it is still possible to gather data on several parameters of command usage on Multics.

2.2 THE MULTICS COMMAND LANGUAGE

The Multics system has a very rich and powerful command language facility. The Honeywell manual contains over 140 regular commands, plus there are several hundred others that reside in the standard search directories and are available to all users. User programs are executed by the same mechanism as the system commands. Almost all the system commands have abbreviations and there is a facility that allows users to specify their own abbreviations. These abbreviations can represent an entire command line and may contain one or more commands separated by semicolons. Multics also has the feature of allowing iterations of the same command with different arguments. For instance, a command followed by a list of arguments enclosed in parentheses would be repeatedly called with the next argument in the list until all were exhausted. It is also possible to invoke a file containing a list of commands to be executed. These command files are known as exec_coms in Multics.

Although such command flexibility is very desirable, it makes command usage monitoring very difficult. The first simplification done in this study was to limit the number of specifically monitored commands to only those defined in the system command language manual and their abbreviations. A list of these commands is included in Appendix A. Any other commands were simply classified together into one category. The second step was to decide how to handle the multiple command constructs. In the case of several commands on one line, each one was counted individually, even though the time between them would be almost insignificant. The reasoning for this was that command usage counts would be more important than think times. For commands that have iteration lists, all the iterations were counted as one invocation of the command. For the command files, the only command that counted was the one that executed the command file. The only exception to this was the command file that is executed automatically when the user logs in, known as the start_up.ec.

2.3 HOW THE MEASUREMENTS WERE TAKEN

Since the system itself could not be modified, the programs that did the monitoring were essentially copies of the system's command processors with the necessary calls to the data collection program. By placing a call to initialize the monitoring in the user's start_up.ec, every time the user logged in, the links to the system command processor were replaced with links to the monitoring package. During the initialization, a file was created that contained all the data for that user's session. For each of the commands being monitored, the following parameters were measured:

1.) The number of times the command was executed.

   The commands are recognized in the command processor by name. For instance, if a user has his own program called "print", then the monitor will classify it as the system command. This should not occur too often since the user may not like the ambiguity involved.

2.) The time since the last command finished, also known as think time.

   Precisely, this time interval has assumed to start as soon as the previous command had finished execution, but before the ready message was printed. (1) The end of the interval was assumed to occur when the command processor recognized that it had received a valid command line and

---

(1) The ready message is usually a 25 character string printed after each command. This can easily be changed by the user.

not a null one. (2) The times were read off of the system clock at the two time points and the difference was summed up along with the sum of the square for the purposes of obtaining the mean and the variance. Also, in order to get an idea of the distribution of think times, a histogram with intervals of one second kept frequency counts of the think time values.

3.) The amount of resource usage by the command.

This was measured by calling a Multics subroutine which returns values for the CPU time, number of page faults, and memory units (3) used since the time the user's process was created. This subroutine was called after each command finished and had returned to the command processor. The difference between the calls then indicated the amount of usage by the command, including command processor and monitoring overhead. These differences and the sum of their squares were kept for each command that was monitored.

For the entire logon session, the following information was recorded: time of login and logout, number of processes created, number of times the break or quit button was used, the number of users at logon time, the total number of

---

(2) An example of a null command line is a line containing blanks, tabs, or semicolons, but no alphabetic characters.

(3) The memory unit is a unit of measure used for accounting on Multics and is used as an estimate for the user's working set.

commands used, and the total resources used during the session.

To get an idea how much overhead was attributable to the monitoring, some test scripts were run as monitored absentee jobs and compared with runs that were not monitored. The overhead of initializing the monitoring programs ranged from about 0.7 to 1.0 seconds of CPU time, depending on the system load. The overhead per command was determined to be about one or two milliseconds. The absolute minimum CPU time necessary to execute a command on the normal system was 20 milliseconds, as reported by the standard ready message. This was measured by invoking a program which simply returned and making sure that no page faults were occuring. The monitoring program may have also changed the paging behavior to some degree, but these effects were not fully investigated.

Several times each day, an absentee job would collect the individual users' session records, print them out (4), accumulate them by the users' project names, and delete them to save space. The totals of each of the project users' session records would thus be stored in the project records. The disadvantage to this is that the data on individual users were assimilated into the project totals and could only be studied by examining the print-outs. Therefore, this study primarily examines the project and over-all totals, rather

---

(4) At this point it would have been desirable to write the records onto a tape for future study, but tape drives were not available.

than looking in detail at specific types of sessions.

2.4 DESCRIPTION OF MONITORED PROJECTS

For the monitoring process to begin, each user has to place a command in his start_up.ec; this would have meant asking for the cooperation of a large number of users. Fortunately, in Multics, a project administrator has the option of specifying a start_up.ec for his entire project. Two courses in the department that were doing work on Multics agreed to participate in the study. These two courses shall be refered to as 6-030 and 6-176 since these are also their project names. The 6-030 project consisted of about 97 students learning PL/1 programming concepts and writing small programs as their assignments. The course assumed no prior programming experience, so these users could be considered first-time or novice time-sharing users. The students in the 6-176 project were taking a laboratory course in computer system performance measurement and evaluation. Their assignments consisted of having to write PL/1 programs to study such things as program performance and system modeling. There were about fifteen users in this project and some programming experience had been assumed. These users could be considered as intermediate-level student time-sharing users.

The third project monitored was a combination of the CompSys and CSR projects. These two projects were merged since they essentially have similar users and some of them have accounts in both projects. The users in these projects were mostly graduate and undergraduate students (including the author) and two secretaries, all in the Computer Systems Research (CSR) division of the Laboratory for Computer Science. There was a wide variety of work, ranging from text editing and manuscript formatting to developing system programs. For the most part, these users were fairly sophisticated and had a good knowledge of the internals and externals of the Multics system. There were about eight users in these two combined projects.

One of the problems of having to seek participation to be monitored was that not everyone was monitored during the same period or for the same amount of time. This meant that such things as login frequency by project could not be compared. It also means that some of the results might be skewed in favor of one project, but the discussion of the results will try to point any such tendencies out.

III - DISCUSSION OF RESULTS

3.1 PROJECT TOTALS

The three projects were monitored over periods ranging from two weeks to four weeks. Since the number of users and use of the system varied so much between projects, it was not possible to get unbiased data that would be generalizable to all three projects. However, it is possible to analyze the data within projects and compare the averages with those from the other projects. A summary of these results can be seen in Table 1 below. Printouts of the details are included in Appendix B.

TABLE 1

Summary of Results by Project
(all times in seconds)

Project	6-030	6-176	CompSys-CSR
No. of sessions	324	99	140
Commands monitored	6,755	3,401	5,435
Command vocabulary	46	52	93
Think time / command	39.4	37.2	22.8
Averages per session:			
Logon time	3,804	2,831	2,175
No. of commands	20.8	34.4	38.9
CPU time	20.9	26.0	31.8
Page faults	4,339	5,204	9,394

As would be expected, the average think time between commands varied substantially between beginners and experienced users of Multics. The average CompSys-CSR user had shorter logon sessions, used more commands and CPU time, and had more page faults than the other two projects. The row entitled "Command vocabulary" shows the size of the subset of monitored commands that were ever used by the project. This gives an indication of the experience or variety of work done by the users in the project. The CompSys-CSK users, for instance, used over two-thirds of the set of standard commands, while the other two projects used a little over one-third of the set.

The details of the command usage for each project are included in Tables 2, 3, and 4. These tables show the fifteen most frequently used commands in each of the projects. These were summarized from Appendix B. The column marked "Percent of total" is the number of times a command was used by the project divided by the total number of commands. The column titled "Times/session" is the number of times a command was used divided by the number of sessions. This gives an indication of how frequently a command was used in a session. The fourth column shows the average amount of time the user thought and typed before the command was executed. The average think times that are below one second belong to those commands that are usually executed in the start_up.ec. The monitored commands include, besides the commands selected from the list in Appendix A, two special categories. The one labeled "not_monitored" includes the valid commands that executed either user programs or any commands which were not explicitly monitored. The second category is called "not_found". These are commands typed by the user that were not found in the search directories. This indicates a type of user error and is usually in the form of mispelling or mistyping a command or is a case of mistaken context, such as an editor command given at the system command level or trying to execute a program that is not contained in the working directory.

The five most frequently used commands by the 6-030 project, shown in Table 2, account for 62% of the command usage in the project. This seems to indicate that the user would create and edit a PL/1 program using edm (1) , format the source program using the indent command, compile it, execute it a couple of times if the compilation was successful, and then print out the working program. This type of behavior is expected since the users were primarily doing their PL/1 assignments on the system.

The results for the 6-176 project, as shown in Table 3, are not as easy to analyze. This seems to be a result of two factors: First, the assignments for this course were not as rigidly defined as those in the previous project and thus, different patterns of work would be expected. The second factor is that the users in this project used, on the average, more commands per session than the 6-030 project, as it is shown in Table 1. However, the command vocabulary (number of different commands used) did not differ significantly between

---

(1) A simple context line editor.

TABLE 2

Command Usage Data
for the 6-030 Project
(top 15 commands)

Number of sessions = 324
Number of commands = 6,755

Command	Count	Percent of total	Times/ session	Think time
not_monitored	1256	18.59	3.88	20.30 (sec)
edm	1122	16.61	3.46	93.43
pl1	691	10.23	2.13	22.17
indent	583	8.63	1.80	24.01
print	560	8.29	1.73	43.60
add_search_rules	325	4.81	1.00	0.11
release	313	4.63	0.97	21.39
set_tty	301	4.46	0.93	10.13
not_found	289	4.28	0.89	59.39
logout	276	4.09	0.85	62.15
delete	271	4.01	0.84	24.17
list	254	3.76	0.78	29.62
probe	91	1.35	0.28	37.29
unlink	69	1.02	0.21	22.23
copy	60	0.89	0.19	78.82

TABLE 3

Command Usage Data
for the 6-176 Project
(top 15 commands)

Number of sessions = 99
Number of commands = 3,401

Command	Count	Percent of total	Times/ session	Think time
not_monitored	705	20.73	7.12	40.75 (sec)
delete	311	9.14	3.14	37.16
list	242	7.12	2.44	34.36
exec_com	241	7.09	2.43	24.14
print	198	5.82	2.00	37.23
release	182	5.35	1.84	19.46
edm	150	4.41	1.52	63.66
copy	148	4.35	1.49	39.61
pl1	132	3.88	1.33	27.83
not_found	125	3.68	1.26	62.35
mail	116	3.41	1.17	4.54
accept_messages	90	2.65	0.91	0.11
logout	89	2.62	0.90	66.16
change_wdir	74	2.18	0.75	33.12
how_many_users	73	2.15	0.74	40.79

the two projects. It seems that the 6-176 users used more commands during a session than the 6-030 users, but still limited themselves to using less than half the set of standard commands.

In Table 4 , the results for the CompSys-CSR projects show that the non-standard commands and/or user programs represent a greater percentage of the used commands as compared to the other two projects. Furthermore, this project used more than twice the number of the standard commands than the 6-030 project. With such diversity of command usage it is difficult to characterize the behavior of the users in this project, but some patterns of use do emerge. For instance, many of these sophisticated users have several directories scattered throughout the Multics file system hierarchy, which may explain the high usage of the commands to list a directory and change the working directory. This usage could be the result of such things as forgetting which files are in a particular directory and having to look for them. The "not_found" category is the fourth command on the list and makes it seem as if more command errors occur in this project than in the others. One possible reason is that these users may forget which directory they are working in and try to execute programs that are not in the directory. However, the main reason the not_found category is ranked so high is that the usage was spread out over more of the other commands. In fact, looking at all three projects, the percentages of the

TABLE 4

Command Usage Data
for the CompSys-CSR Project
(top 15 commands)

Number of sessions = 140
Number of commands = 5,435

Command	Count	Percent of total	Times/ session	Think time
not_monitored	1547	28.46	11.05	24.11 (sec)
list	515	9.48	3.68	19.64
change_wdir	381	7.01	2.72	23.70
not_found	236	4.34	1.69	15.73
delete	233	4.29	1.66	27.28
archive	170	3.13	1.21	28.79
accept_messages	128	2.36	0.91	0.18
who	123	2.26	0.88	35.57
logout	109	2.01	0.78	74.80
print	102	1.88	0.73	31.28
release	99	1.82	0.71	12.30
edm	97	1.78	0.69	38.17
qedx	95	1.75	0.68	38.23
dprint	92	1.69	0.66	20.60
abbrev	81	1.49	0.58	5.61

not_found occurences differ by at most 0.66 percent. This suggests that the rate of typing bad commands is around 4 percent of all commands typed into the system.

The data for all the projects was combined into one total and the results are included in Appendix C. As mentioned earlier, it is not known to what extent any of the projects may have skewed the data. Table 5 shows a summary of the command usage for all the projects combined. A fifth column was added which shows if any particular project contributed a large percentage to the totals for that command. Thus, the edm command may appear as the second most used command in all three projects, but 82% of its usage was due to the 6-030 users.

It is interesting to note in Table 5 that two commands have significantly longer think times than the others. The longest is the edm command and the other is the logout command. The long think time before a logout command can be explained because most users pause trying remember if they have done everything they wanted to do in the session before committing themselves to the logout. In his study on the TSS system, Boies (1974, p. 16), reported similar logout behavior. The long think time before using the editor is harder to explain. Perhaps what is happening is that the user is using the editor to correct mistakes in his program, but before that can be done, the mistake must be found. In some cases, the user may even have to refer to a manual in order to correct

TABLE 5

Command Usage Data
for all Three Projects
(top 15 commands)

Number of sessions = 563
Number of commands = 15,591

Command	Count	Percent of total	Times/ session	Think time	Group attribution
not_monitored	3508	22.80	6.23	26.09	
edm	1369	8.78	2.43	86.26	82%-I
list	1011	6.48	1.80	25.67	
pl1	876	5.62	1.56	22.44	79%-I
print	860	5.52	1.53	40.68	
delete	815	5.23	1.45	30.02	
not_found	650	4.17	1.15	44.11	
indent	622	3.99	1.10	23.61	94%-I
release	594	3.81	1.06	19.28	
change_wdir	488	3.13	0.87	26.16	78%-III
logout	474	3.04	0.84	65.81	
add_search_rules	349	2.24	0.62	0.45	93%-I
set_tty	345	2.21	0.61	12.12	87%-I
exec_com	259	1.66	0.46	24.46	93%-II
copy	238	1.53	0.42	46.17	

Group I - 6-030 Project
Group II - 6-176 Project
Group III - CompSys & CSR Projects

the mistake.

By averaging all the data that was collected, it is possible to create a user model that is a composite of all the users in the three projects. This composite session is described in Table 6. Similarly, by averaging the results of the per-command data, it is possible to describe the composite command, as in Table 7. It is important to understand that these composite user models may not be generalizable to the entire population of Multics users. Rather, this model describes the subset of users that were monitored. A study involving all users of Multics would be necessary to claim that the model can be generalized. One test to measure the validity of the composite command is to divide the CPU time by the number of page faults. This should result in the mean time between page faults (mtbpf). The composite predicts a mtbpf of 4.26 milliseconds. This is close to the actual metered mtbpf on Multics, which is in the range of 4 to 7 milliseconds, including system overhead.

TABLE 6

The Composite Multics Session

(based on 563 sessions)

The composite user logs on when there are 29.25 users
    on the system.
He logs out after 53 minutes 48 seconds.
Executes 27.7 commands and thinks 33.2 seconds between them.
Does 1.44 quits.
Uses the following resources:
    25.50 CPU seconds
    440.66 Memory units
    5,748 page faults

TABLE 7

The Composite Multics Command

(based on 15,591 commands)

0.792	CPU seconds
80.146	Seconds of real time
186	Page faults
13.665	Memory units

3.2 THINK TIME DISTRIBUTION

While gathering data on command usage, the monitoring program also kept a histogram of think times in order to get an idea of the think time distribution. The histogram had intervals of one second and recorded values of think times up to twenty minutes. This tool accumulated the results for all monitored users rather than keeping separate distributions for each project. A plot connecting the midpoints of the histogram can be seen in Figure 1. The impulse is hypothesized to exist because of the large number of think times that were from zero up to one second. This low value occurs when: commands are executed in the start_up.ec; there is more than one command on a line; or the user types ahead on a full-duplex terminal. This impulse would occur at about 0.1 seconds since this is the overhead for the command processor to look at the next command. The next peak occurs around 8 seconds and represents the maximum rate of user interaction. This is the time the user spends typing a short command as soon as the previous command has finished. After that peak, the distribution falls off rather quickly, but has a long tail as it is shown in the cumulative plot in Figure 2. Almost 90 percent of the think times were less than 70 seconds long.

The most popular probability model for user think times in analytical and simulation studies has been the exponential distribution (Sekino, 1972, p.34). Its features are that it simplifies the analysis of queuing models. To test the validity of such a think time model for the Multics system, the plots for an exponential distribution with a mean the same as the sample mean, are shown in Figures 1 and 2 for comparison with the sample distribution. The exponential distribution is a rather poor fit and does not seem adequate to characterize the think time. The sample distribution seems to be composed of three parts: the impulse near zero; the peak near 8 seconds; and the long tail. The three types of behavior that may explain these parts are: when the user is interacting faster than the system; when the user and system

FIGURE 1

Probability Distribution

of User Think Times

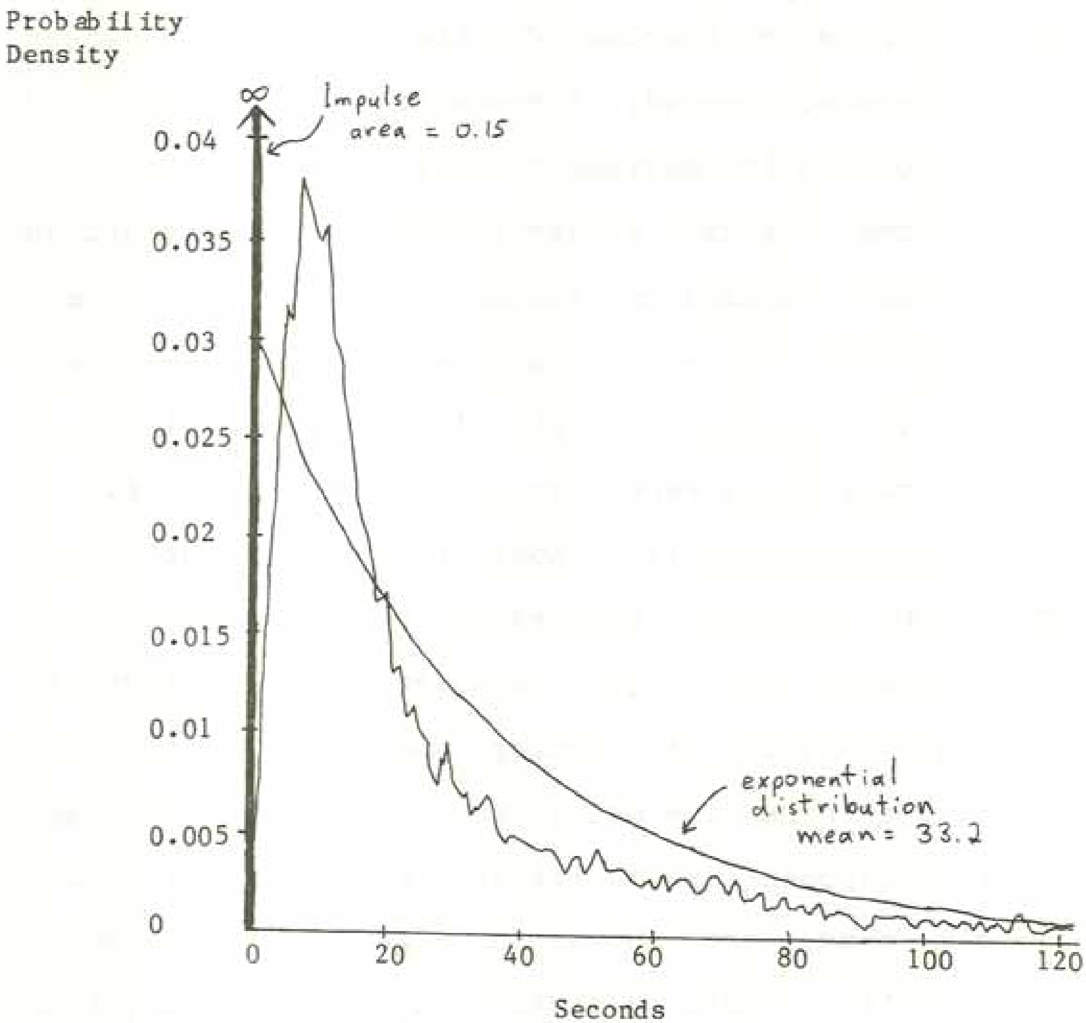
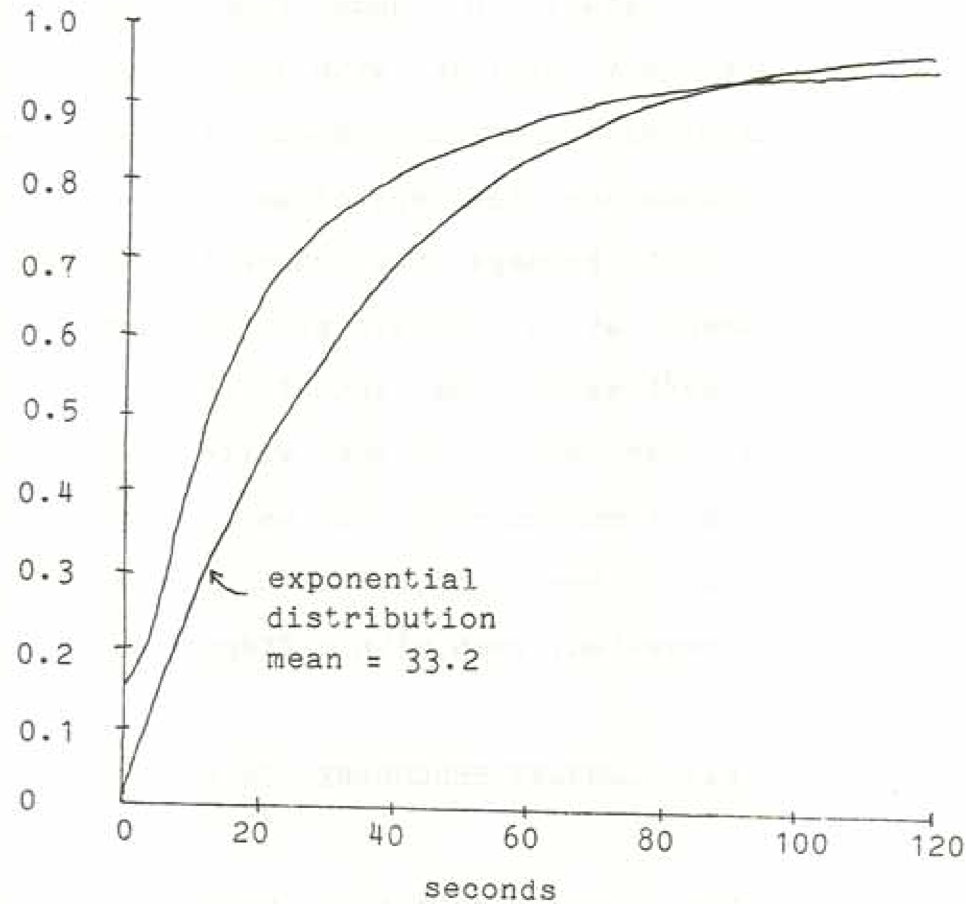


Figure 2
Cumulative Distribution
of User Think Times



interact one after another; and when the user pauses for an indeterminate period, such as to look something up or leave the terminal for a cup of coffee.

Before the Multics system existed at M.I.T., there was the CTSS timesharing system and a study of the think times of CTSS users was done by Scherr (1965). Figures 3 and 4 compare the results of this study to those from CTSS. The two distributions are very similar with approximately the same area under the impulses and peaks in about the same place. The mean and median values are also very close. This suggests the powerful notion that perhaps the characteristics of a population of users at a certain site may not change very much, even if the systems or the actual people do change. This would make user models a more valid concept and lend support to the idea of benchmarking. It would be interesting to test this notion by seeing if the distribution obtained in this study can be repeated, even with different projects.

3.3 COMPARISONS WITH CURRENT BENCHMARK SCRIPTS

The performance of new versions of the Multics system are evaluated by performing a benchmark test called the acceptance test. This test puts a workload on the system by simulating several concurrent users. The performance is measured by reading the various system meters and seeing how long the test actually takes. The simulated users are absentee jobs that

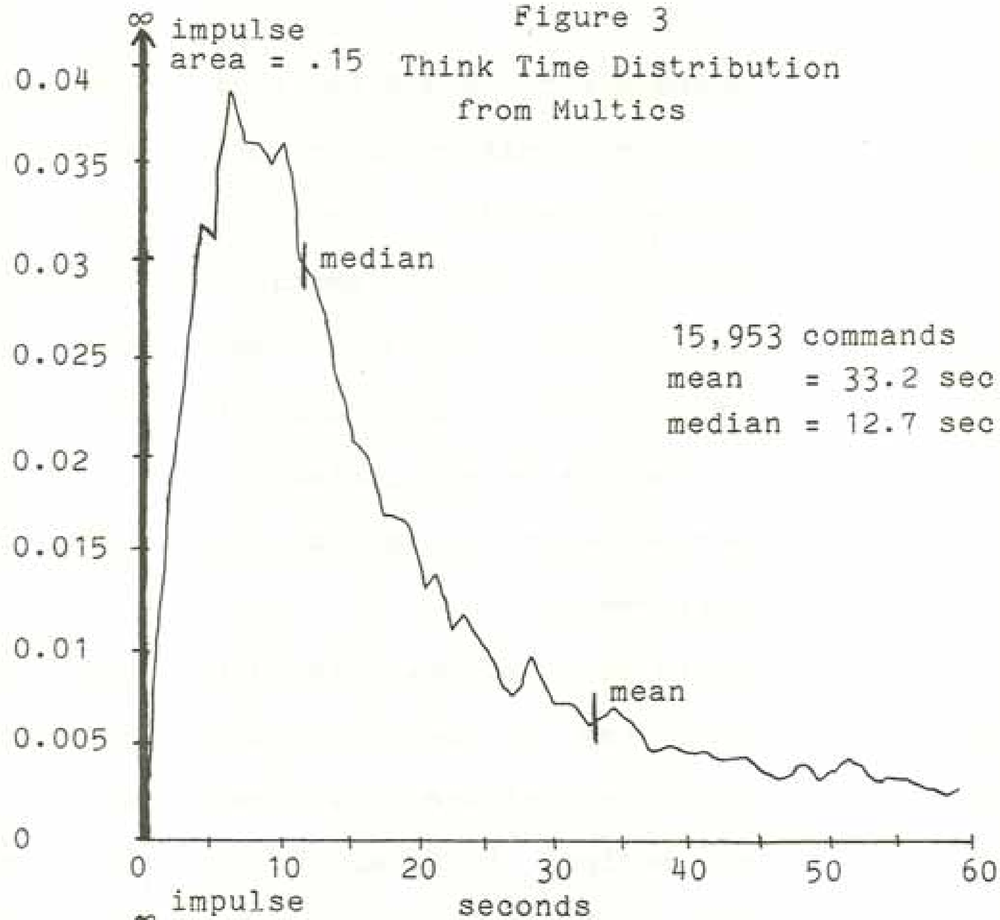


Figure 3
Think Time Distribution
from Multics

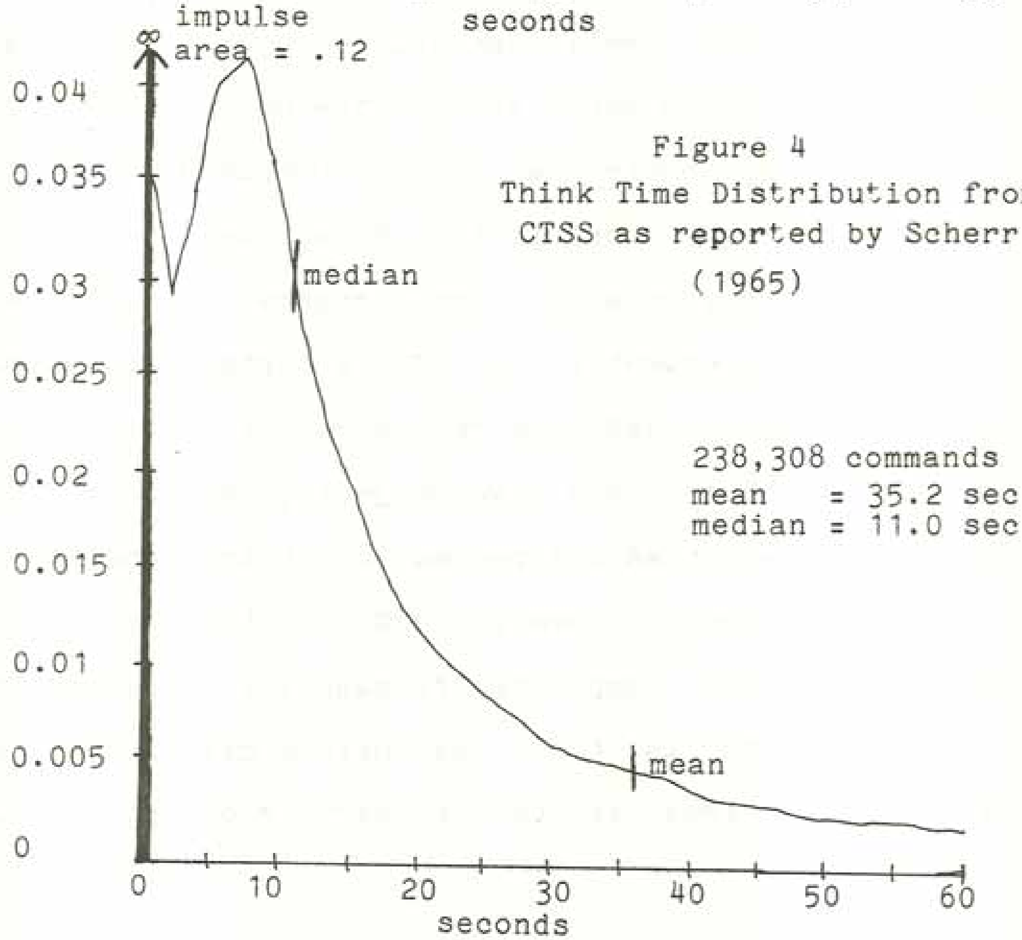


Figure 4
Think Time Distribution from
CTSS as reported by Scherr
(1965)

follow a set of command scripts and the think times are random numbers generated from a uniform distribution on the interval from two to fifteen seconds. There are presently five scripts, four of which are almost identical and contain about 10 compile commands, 4 edit commands, and about 21 other commands. The fifth script was designed to create heavy paging activity in the system and contains about 20 compile commands interspersed with a special command that flusnes pages out of primary memory.

One of the goals of a benchmark is to simulate a normal workload in the system in order to observe how the system behaves under normal circumstances. The major problem is in selecting command scripts that model the work done by the users of the system. (Hellerman and Conroy, 1975) One solution would be to conduct a study such as the one in this paper to help design a reasonable user model. Compared to the findings in this paper, the current scripts do not seem to characterize the Multics users very well. For instance, the compilers were not used as frequently as the scripts use them. The CompSys-CSR project used more resources per session, on the average, than the other two projects, yet the PL/1 command accounted for less than one percent of their commands. Thus, users who produce a heavy load on the system do not necessarily use the compilers frequently. Another problem with the scripts is that the uniform distribution used to generate think times is not a very good match for the distribution that was found in this study.

A possible method for designing a more accurate benchmark would be to characterize the users by projects, since this study has shown that user behavior does vary between projects. The next step would be to determine how often users of each project log in each day. A set of project user models can then be designed, and the benchmark constructed as a mix of these models. For instance, if half of the number of sessions on the system are from the 6-030 project, then half of the user models should be based on the characteristics of 6-030 users. The major problems with this approach are its generalizability and the possibilty that the percentage of sessions by a project may not be a very stable parameter.

IV - CONCLUSIONS

4.1 - SUGGESTIONS FOR FURTHER RESEARCH

One of the limitations to this study was that the individual session records could not be stored anywhere. This prevented a closer investigation into the behavior of individual users, rather than the project as a whole. It would be interesting to study such individual characteristics as: do users who login frequently have different behaviors from infrequent users? or what percentage of the users login just to execute one or two commands before logging out? Another limitation to this study was that the user behavior within a command, such as an editor, could not be measured. In other words, not all interactions between the system and the user were monitored, only those at the command level. Knowing what a user does within an editor could be very helpful in designing new editors, for instance. Also, the actual think time distribution may be different, if the activities within editors are monitored.

Since the think time histogram worked so well in this study, other histograms should have been kept for other parameters such as CPU time and page faults per command. A useful system metering tool for Multics could be obtained by installing these histogram functions into the system command processors by using the techniques developed for the monitoring program used in this study. Also, the statistics on the command usage for each individual session could be made available to the user so that he could get a better picture of how he uses the system. Hopefully, this paper has shown that gathering data on user behavior can be very useful in the field of system performance measurement and would be worth the effort to build user monitoring capabilities into the system.

LIST OF REFERENCES

Boies, S.J., "User Behavior on an Interactive Computer System," IBM Systems Journal, Vol. 13, No. 1, 1974, pp.2-18.

Hellerman, H. and Conroy, T.F., Computer System Performance, New York: McGraw-Hill, 1975, pp. 246-250.

Roach, R., Multics Technical Bulletin, MTB-126, Unpublished internal memo, 1974.

Scherr, A.L., "An Analysis of Time-Shared Computer Systems," Ph.D. Thesis, Project MAC Report MAC-TR-18, June 1965, pp. 7-11, (Also available from MIT Press, Cambridge, Mass.).

Sekino, A., "Performance Evaluation of Multiprogrammed Time-Shared Computer Systems," Ph.D. Thesis, Project MAC Report MAC-TR-103, Sept. 1972, pp. 201-206.

Treau, S., "Methodology for Interactive Computer Service Measurement and Evaluation," Dept. of Computer Science, Univ. of Pittsburgh, Pittsburgh, Penn., Technical Report 74-2, 1974.

APPENDIX A

List of Monitored Commands

ID	COMMAND NAME	ABBREV
1	not_monitored	
2	abbrev	ab
3	accept_messages	am
4	add_name	an
5	add_search_rules	asr
6	adjust_bit_count	abc
7	answer	
8	apl	apl
9	archive	ac
10	assign_resource	ar
11	basic	
12	basic_system	bs
13	bind	bd
14	calc	calc
15	cancel_abs_request	car
16	cancel_daemon_request	cdr
17	change_default_wdir	cdwd
18	change_error_mode	cem
19	change_wdir	cwd
20	check_info_segs	cis
21	close_file	cf
22	cobol	
23	compare	
24	compare_ascii	cpa
25	console_output	co
26	copy	cp
27	copy_cards	
28	create	cr
29	create_dir	cd
30	cumulative_page_trace	cpt
31	debug	db
32	decode	
33	defer_messages	dm
34	delete	dl
35	delete_acl	da
36	delete_dir	dd
37	delete_force	df
38	delete_iacl_dir	did
39	delete_iacl_seg	dis
40	delete_name	dn
41	delete_search_rules	dsr
42	do	do
43	dprint	dp
44	dpunch	dpn
45	dump_segment	ds
46	edm	edm
47	encode	
48	enter_abs_request	ear
49	exec_com	ec
50	file_output	fo
51	fortran	ft
52	fortran_abs	fa
53	fs_chname	

ID	COMMAND NAME	ABBREV
54	gcos	gc
55	get_com_line	gcl
56	get_quota	gq
57	help	help
58	how_many_users	hmu
59	immediate_messages	im
60	indent	ind
61	initiate	in
62	io_call	io
63	line_length	ll
64	link	lk
65	list	ls
66	list_abs_requests	lar
67	list_acl	la
68	list_daemon_requests	ldr
69	list_iacl_dir	lid
70	list_iacl_seg	lis
71	list_ref_names	lrn
72	list_resources	lr
73	logout	
74	mail	ml
75	memo	memo
76	move	mv
77	move_quota	mq
78	new_proc	
79	page_trace	pgt
80	pl1	pl1
81	pl1_abs	pa
82	print	pr
83	print_attach_table	pat
84	print_auth_names	pan
85	print_default_wdir	pdwd
86	print_messages	pm
87	print_motd	pmot
88	print_proc_auth	ppa
89	print_request_types	prt
90	print_search_rules	psr
91	print_wdir	pwd
92	probe	pb
93	profile	
94	program_interrupt	pi
95	progress	pg
96	qedx	qx
97	ready	rdy
98	ready_off	rdf
99	ready_on	rdn
100	release	rl
101	rename	rn
102	resource_usage	ru
103	run_cobol	rc
104	runoff	rf
105	runoff_abs	rfa
106	safety_sw_off	ssf

ID	COMMAND NAME	ABBREV
107	safety_sw_on	ssn
108	send_message	sm
109	set_acl	sa
110	set_bit_count	sbc
111	set_cc	
112	set_com_line	scl
113	set_incl_dir	sid
114	set_incl_seg	sis
115	set_search_rules	ssr
116	set_tty	stty
117	sort_seg	ss
118	start	sr
119	status	st
120	terminate	tm
121	terminate_segno	tms
122	terminate_refname	tmr
123	terminate_single_refname	tmsr
124	trace	
125	trace_stack	ts
126	truncate	tc
127	unassign_resource	ur
128	unlink	ul
129	walk_subtree	ws
130	where	wh
131	who	who
132	not_found	

APPENDIX B

Detail of Project Totals

(Note - the initials s.d. stand for standard deviation.)

****
REPORT FOR 6-030   PROJECT.
****

Total Number of sessions: 324   Interactive: 324   Absentee: 0
Number of processes: Total: 325   Avg: 1.000
Number of quits: Total: 446   Avg: 1.375
Avg. max load units: 85.000   Avg. load units: 26.000
Average CPU secs per session: 20.882   S.D. 28.163
Average memory units per session: 347.079   S.D.: 265.229
Average page faults per session: 4,339   S.D.: 3,518
Average session length (in secs): 3803.967   S.D.: 4148.903

Number of commands used: 6755   Per session: 20.85   S.D.: 16.35
Average think time: 39.4 secs.

****

Command Name	Code Number	Count	Page Faults (mean)	Page Faults (sd)	Memory Units (k) (mean)	Memory Units (k) (s.d.)	CPU Time (mean)	CPU Time (s.d.)	Real Time (mean)	Real Time (s.d.)	Think Time (mean)	Think Time (s.d.)
not_monitored	1	1256	195	485	14.518	23.611	1.571	10.804	86.603	329.702	20.303	59.987
acurev	2	1	46	00	3.026	0.000	0.044	0.000	0.065	0.000	58.432	0.000
accept_messages	3	3	256	87	30.583	7.080	0.882	0.395	1.903	1.276	33.288	34.997
add_search_rules	5	325	38	22	7.503	3.871	0.208	0.023	0.476	0.399	0.109	0.346
archive	9	10	127	76	9.036	6.054	0.261	0.174	2.269	2.702	39.533	31.145
calc	14	2	297	154	12.869	8.302	0.369	0.226	126.285	37.845	51.570	7.195
change_wdir	19	33	56	30	3.909	3.256	0.100	0.096	0.149	0.141	39.042	35.900
copy	26	60	92	41	6.990	4.712	0.225	0.207	3.912	8.963	78.823	78.211
create	28	1	29	00	0.810	0.000	0.068	0.000	0.136	0.000	17.402	0.000
create_dir	29	4	86	20	8.372	3.360	0.215	0.155	0.370	0.160	46.631	17.810
delete	34	271	60	25	3.556	1.912	0.145	0.061	0.461	0.790	24.177	22.818
delete_acl	35	1	91	00	6.810	0.000	0.146	0.000	0.351	0.000	53.706	0.000
delete_dir	36	1	124	00	7.114	0.000	0.333	0.000	6.737	0.000	6.160	0.000
dprint	43	1	91	00	6.814	0.000	0.103	0.000	0.344	0.000	6.253	0.000
dump_segment	45	13	229	296	5.063	5.831	0.419	0.528	404.048	623.663	54.170	64.921
edm	46	1122	398	543	10.783	14.261	0.549	1.006	666.859	1259.745	93.434	183.575
get_quota	56	1	103	00	10.382	0.000	0.339	0.000	1.122	0.000	13.150	0.000
help	57	36	123	63	6.578	3.604	0.180	0.103	39.558	38.585	48.506	72.021
immediate_messages	59	12	50	17	2.328	1.531	0.306	0.037	0.417	0.115	12.744	9.333
indent	60	583	73	32	4.642	3.170	0.254	0.090	0.707	0.555	24.006	39.171
initiate	61	1	48	00	1.296	0.000	0.079	0.000	0.070	0.000	8.664	0.000
link	64	9	88	38	5.975	4.018	0.098	0.045	21.041	32.303	157.320	165.462
list	65	254	64	37	4.726	4.177	0.137	0.095	0.304	0.330	29.517	71.242
list_acl	67	11	34	17	1.599	1.077	0.078	0.014	0.120	0.049	8.891	6.743
logout	73	276	83	27	6.195	3.172	0.102	0.090	0.525	0.429	62.153	86.555
mail	74	28	177	118	14.124	9.754	0.417	0.319	35.954	89.748	45.082	44.933
move	76	3	88	19	7.484	3.037	0.339	0.481	0.639	0.765	46.237	30.317
new_proc	78	1	254	00	41.988	0.000	1.490	0.000	4.579	0.000	13.271	0.000
pl1	80	691	422	324	65.068	38.431	3.657	2.626	12.316	20.514	22.174	32.304
pl1_abs	81	4	136	93	13.522	12.360	0.343	0.366	1.440	1.697	26.711	16.640
print	82	560	103	60	4.659	3.153	0.440	0.326	83.475	92.329	43.603	65.533
print_wdir	91	15	58	44	2.565	2.155	0.131	0.144	0.256	0.379	135.134	282.937
probe	92	91	215	158	13.265	10.362	0.380	0.331	145.769	183.466	37.290	51.515
program_interrupt	94	12	209	271	10.222	12.224	0.302	0.341	109.560	167.423	12.370	16.333
qedx	96	2	251	146	10.727	5.901	1.018	0.428	105.324	58.451	27.895	9.043

release	100	313	28	24	2.025	2.300	0.044	0.042	0.010	0.033	21.391	36.900
rename	101	12	56	29	3.019	1.920	0.094	0.042	0.411	0.544	73.677	119.535
resource_usage	102	10	53	21	3.284	2.094	0.037	0.024	0.372	0.230	34.786	25.331
send_message	108	2	155	98	12.174	8.304	0.242	0.237	27.756	38.547	34.518	2.479
set_acl	109	25	61	24	4.831	5.109	0.147	0.143	0.347	0.432	133.781	412.059
set_iacl_seg	114	2	40	07	1.981	0.213	0.071	0.015	0.106	0.356	18.395	5.070
set_tty	116	301	139	62	19.042	5.659	0.494	0.067	0.352	0.339	10.133	5.790
start	118	21	171	216	7.911	7.552	1.887	4.067	150.411	345.609	26.087	33.119
status	119	3	52	17	3.032	1.586	0.092	0.018	0.392	0.085	43.635	14.135
unlink	128	69	05	05	0.679	0.407	0.061	0.005	0.085	0.045	0.004	0.027
where	130	4	82	67	3.954	3.589	0.126	0.033	0.677	1.064	42.195	45.472
who	131	10	47	15	2.368	1.054	0.149	0.049	0.363	0.171	17.900	15.525
not_found	132	249	68	26	4.317	2.633	0.093	0.062	0.382	0.434	59.389	103.157
TOTAL RESULTS	0	3755	189	354	14.489	24.677	0.912	4.886	139.141	585.788	39.409	93.973

00044

****
REPORT FOR 6-176 PROJECT.
****

Total Number of sessions: 99 Interactive: 99 Absentee: 0
Number of processes: Total: 106 Avg: 1.063
Number of quits: Total: 234 Avg: 2.313
Avg. max load units: 83.688 Avg. load units: 28.625
Average CPU secs per session: 26.025 S.D. 33.794
Average memory units per session: 397.529 S.D.: 330.384
Average page faults per session: 5,204 S.D.: 5,250
Average session length (in secs): 2831.399 S.D.: 3109.156

Number of commands used: 3401 Per session: 34.35 S.D.: 33.43
Average think time: 37.2 secs.

****

Command Name	Code Number	Count	Page Faults (mean)	Page Faults (sd)	Memory Units (k) (mean)	Memory Units (k) (s.d.)	CPU Time (mean)	CPU Time (s.d.)	Real Time (mean)	Real Time (s.d.)	Think Time (mean)	Think Time (s.d.)
not_monitored	1	705	160	327	9.878	23.706	1.452	3.933	45.147	151.123	40.752	87.489
abbrev	2	18	36	26	1.824	1.958	0.070	0.050	0.075	0.040	17.916	21.659
accept_messages	3	90	53	42	6.200	2.205	0.238	0.046	1.360	6.295	0.114	0.177
add_name	4	1	59	00	3.350	0.000	0.149	0.000	0.306	0.000	122.248	0.000
add_search_rules	5	4	57	27	3.827	2.447	0.162	0.025	0.377	0.169	27.147	11.134
archive	9	5	83	11	7.015	1.858	0.176	0.111	0.510	0.529	82.927	51.465
basic	11	26	621	645	26.924	23.756	3.239	4.139	408.601	448.025	21.647	35.037
calc	14	2	403	435	12.495	12.449	0.380	0.366	634.104	792.967	823.638	823.621
change_wdir	19	74	43	24	2.112	1.913	0.056	0.043	0.239	0.292	33.124	30.793
close_file	21	12	58	35	3.769	2.538	0.086	0.050	0.446	0.588	15.738	18.188
copy	26	148	73	46	5.263	4.370	0.194	0.140	0.878	1.970	39.612	31.935
delete	34	311	65	28	3.639	2.085	0.185	0.112	0.584	0.574	37.158	38.157
delete_name	40	2	45	18	2.274	1.510	0.044	0.014	0.144	0.135	38.477	15.395
dprint	43	6	101	55	10.284	5.844	0.280	0.160	0.759	0.613	84.365	104.191
edm	46	150	412	451	11.763	11.050	0.687	0.724	385.992	605.590	53.663	128.052
exec_com	49	241	171	221	17.398	20.262	0.385	0.409	27.405	120.033	24.137	38.954
help	57	17	203	105	10.684	5.173	0.338	0.168	84.139	63.813	34.270	33.725
how_many_users	58	73	33	19	1.513	1.090	0.067	0.045	0.243	0.314	40.788	52.014
indent	60	27	80	44	5.698	3.923	0.274	0.168	0.913	0.735	22.300	11.915
line_length	63	8	51	37	2.322	2.126	0.081	0.036	0.313	0.455	40.647	67.203
link	64	54	88	55	5.238	3.953	0.167	0.137	4.846	8.292	48.323	52.079
list	65	242	50	29	2.910	2.410	0.169	0.079	6.381	0.442	34.355	95.459
list_acl	67	5	59	33	3.196	1.895	0.131	0.051	0.505	0.507	19.061	5.734
list_daemon_reques	68	2	21	21	0.756	0.919	0.165	0.065	0.294	0.243	12.429	3.353
list_iacl_seg	70	1	69	00	4.430	0.000	0.121	0.000	0.781	0.000	15.689	0.000
list_resources	72	2	105	38	9.699	5.163	0.372	0.364	1.161	0.409	46.295	15.959
logout	73	89	82	41	5.486	3.056	0.106	0.077	0.597	0.597	66.156	130.453
mail	74	116	149	83	24.233	13.358	0.620	0.229	11.941	25.995	4.540	19.425
new_proc	78	7	389	191	35.637	13.157	1.812	0.603	12.763	12.110	96.212	142.359
page_trace	79	1	197	00	17.338	0.000	1.131	0.000	110.546	0.000	173.064	0.000
pl1	80	132	554	526	68.156	35.876	4.204	3.468	15.739	25.104	27.831	45.042
print	82	198	93	135	3.286	3.307	0.418	0.932	84.269	257.222	37.232	36.033
print_auth_names	84	1	55	00	2.580	0.000	0.162	0.000	0.246	0.000	12.139	0.000
print_messages	86	1	20	00	0.420	0.000	0.033	0.000	0.096	0.000	42.028	0.000
print_search_rules	90	1	11	00	0.132	0.000	0.073	0.000	0.082	0.000	31.473	0.000

print_wdir	91	4	45	08	2.534	1.409	0.061	0.024	0.127	0.059	47.109	64.551
probe	92	6	363	334	24.728	22.998	0.600	0.621	94.894	99.963	42.407	25.291
profile	93	29	82	48	3.734	2.010	0.476	0.647	77.027	128.199	22.700	19.332
qedx	96	24	164	98	6.433	5.831	0.362	0.248	179.254	167.600	64.246	78.737
release	100	182	32	29	2.071	2.519	0.065	0.063	0.011	0.035	19.458	55.229
rename	101	14	63	32	3.709	2.959	0.127	0.108	1.395	3.019	27.690	9.540
resource_usage	102	43	43	29	2.355	2.060	0.117	0.059	0.273	0.239	30.743	49.341
runoff	104	17	164	106	15.423	10.264	1.748	1.845	112.988	214.374	32.057	24.473
send_message	108	19	129	111	9.581	10.547	0.287	0.279	89.448	242.955	63.547	44.740
set_acl	109	45	61	15	3.794	1.764	0.161	0.159	0.359	0.291	48.545	33.503
set_iacl_seg	114	4	57	35	4.402	3.867	0.167	0.133	0.319	0.292	23.360	17.038
set_tty	116	36	50	34	2.376	2.650	0.092	0.084	0.164	0.170	26.253	17.533
start	118	32	109	155	5.151	5.270	0.268	0.386	37.425	111.278	18.192	21.913
status	119	1	112	00	6.812	0.000	0.325	0.000	0.328	0.000	648.195	0.000
unlink	128	15	51	22	2.495	1.254	0.102	0.046	0.314	0.303	22.233	10.245
who	131	33	53	33	2.915	2.848	0.183	0.102	0.345	0.386	74.029	109.573
not_found	132	125	63	37	3.673	2.557	0.119	0.099	0.460	0.600	62.346	171.719
TOTAL RESULTS	0	3401	136	258	9.889	19.701	0.690	2.174	42.249	189.859	37.244	80.410

00046

****
REPORT FOR CompSys-CSR   PROJECT.
****

Total number of sessions: 140   Interactive: 140   Absentee: 0
Number of processes: Total: 143   Avg: 1.003
Number of quits: Total: 147   Avg: 1.000
Avg. max load units: 83.313   Avg. load units: 37.313
Average CPU secs per session: 31.790   S.D. 50.408
Average memory units per session: 687.725   S.D.: 893.147
Average page faults per session: 9,394   S.D.: 11,550
Average session length (in secs): 2175.206   S.D.: 2459.931

Number of commands used: 5435   Per session: 38.82   S.D.: 40.98
Average think time: 22.8 secs.

****

Command Name	Code Number	Count	Page Faults (mean)	Page Faults (sd)	Memory Units (k) (mean)	Memory Units (k) (s.d.)	CPU Time (mean)	CPU Time (s.d.)	Real Time (mean)	Real Time (s.d.)	Think Time (mean)	Think Time (s.d.)
not_monitored	1	1547	251	928	19.647	92.031	0.335	6.974	26.716	214.500	24.106	88.159
abbrev	2	81	33	33	2.204	2.353	0.067	0.073	0.080	0.036	5.607	10.542
accept_messages	3	128	106	78	10.537	7.155	0.316	0.174	2.160	5.574	0.179	0.147
add_name	4	25	121	62	7.989	4.466	0.429	0.522	4.057	9.333	16.622	13.113
add_search_rules	5	20	87	36	7.313	3.072	0.331	0.028	0.621	0.251	0.591	0.271
adjust_bit_count	6	8	80	30	6.284	4.815	0.204	0.138	0.575	0.432	12.779	5.049
answer	7	2	259	178	16.294	11.842	0.558	0.240	1.753	1.041	26.220	35.222
archive	9	170	217	144	19.367	19.536	0.500	0.281	2.296	2.233	28.790	75.135
assign_resource	10	1	65	00	4.290	0.000	0.124	0.000	0.452	0.000	2.455	0.000
basic_system	12	1	147	00	7.909	0.000	0.442	0.000	10.533	0.000	6.684	0.000
bind	13	35	410	288	43.282	28.130	4.131	3.146	10.818	11.434	17.199	12.472
calc	14	2	55	03	2.854	0.299	0.069	0.023	0.396	0.525	5.762	2.551
cancel_abs_request	15	14	185	118	9.827	5.247	0.390	0.314	1.212	0.995	16.797	9.351
cancel_daemon_requ	16	8	113	49	7.321	2.271	0.215	0.106	0.789	0.904	19.957	8.747
change_error_mode	18	46	29	16	2.846	1.123	0.031	0.017	0.163	0.127	0.369	1.135
change_wdir	19	391	77	60	4.480	3.276	0.126	0.117	0.315	0.333	23.696	45.443
check_info_segs	20	31	242	112	25.612	10.315	1.973	0.405	6.733	4.432	11.356	17.124
close_file	21	1	11	00	0.132	0.000	0.100	0.000	0.251	0.000	1.857	0.000
compare_ascii	24	29	165	81	9.549	5.784	0.399	0.282	8.885	37.913	24.594	28.259
console_output	25	53	32	26	2.461	1.792	0.046	0.026	0.219	0.533	0.573	1.252
copy	26	30	203	81	21.235	9.189	0.477	0.155	2.065	1.551	13.178	5.625
create	28	5	86	46	4.467	2.706	0.186	0.094	0.675	0.409	15.771	14.012
create_dir	29	6	82	34	4.974	1.587	0.152	0.044	0.562	0.314	16.896	9.517
debug	31	5	595	242	30.742	12.320	1.167	0.598	208.256	405.313	49.312	71.357
delete	34	233	112	59	6.130	3.226	0.295	0.159	0.939	1.323	27.277	73.734
delete_acl	35	8	117	62	7.254	3.236	0.240	0.118	1.062	0.717	44.572	74.376
delete_dir	36	3	285	94	16.791	1.944	1.158	0.896	7.504	2.771	6.910	2.177
delete_force	37	1	14	00	0.210	0.000	0.073	0.000	0.061	0.000	3.579	0.000
delete_incl_dir	38	1	124	00	6.152	0.000	0.121	0.000	2.269	0.000	52.402	0.000
delete_name	40	7	73	28	5.274	2.907	0.142	0.031	0.485	0.676	23.666	13.455
delete_search_rule	41	1	53	00	3.351	0.000	0.175	0.000	0.329	0.000	5.895	0.000
do	42	18	303	171	29.313	14.462	0.631	0.397	12.808	5.443	51.723	115.793
dprint	43	92	202	150	16.093	10.015	0.459	0.330	2.435	2.912	20.596	13.417
dump_segment	45	20	135	95	8.800	5.599	0.674	0.800	6.296	14.179	37.870	43.515
edm	46	97	525	825	19.472	27.297	1.898	6.077	296.593	521.135	38.172	60.343
enter_abs_request	48	15	210	87	12.086	4.434	0.372	0.187	2.223	1.437	37.608	34.732
exec_com	49	18	407	412	38.565	39.265	1.507	1.729	8.239	8.513	28.772	45.850

file_output	50	56	146	76	9.047	4.455	0.280	0.175	0.814	0.921	37.541	32.393
fs_chname	53	2	86	54	4.723	2.945	0.189	0.096	0.139	0.133	31.508	23.555
get_quota	56	77	132	108	8.951	6.151	0.277	0.254	1.287	1.175	21.021	51.059
help	57	81	217	205	11.996	10.634	0.449	0.503	47.213	87.155	19.055	22.737
how_many_users	58	9	184	137	9.385	5.193	0.288	0.197	1.199	0.536	74.537	81.932
immediate_messages	59	1	157	00	6.395	0.000	0.249	0.000	1.365	0.000	49.703	0.000
indent	60	12	112	92	7.175	5.720	0.491	0.262	1.122	0.919	7.501	3.227
initiate	61	8	70	19	4.935	2.063	0.114	0.052	0.184	0.134	9.986	5.174
io_call	62	5	104	31	5.692	2.197	0.181	0.049	0.244	0.154	30.172	21.002
line_length	63	8	60	35	3.509	2.137	0.136	0.067	0.319	0.422	7.345	4.573
link	64	21	142	106	8.893	6.104	0.279	0.252	1.324	2.652	29.334	20.333
list	65	515	94	81	6.140	5.312	0.305	0.391	2.900	29.334	19.644	55.153
list_abs_requests	66	30	182	68	13.229	3.645	0.370	0.135	1.799	1.355	17.511	13.022
list_acl	67	34	100	47	6.534	3.477	0.250	0.126	1.085	1.147	24.550	51.455
list_daemon_reques	68	36	133	67	10.710	5.618	0.320	0.201	1.223	1.261	41.580	102.215
list_iacl_dir	69	4	85	47	4.745	2.916	0.122	0.043	0.923	1.013	14.064	14.212
list_iacl_seg	70	5	61	39	3.760	3.100	0.116	0.048	3.582	0.577	11.276	11.233
list_ref_names	71	6	53	12	3.415	1.341	0.073	0.029	0.215	0.193	27.434	10.752
logout	73	109	119	67	8.518	4.539	0.138	0.091	0.901	0.807	74.797	204.372
mail	74	66	226	133	20.592	10.667	0.639	0.319	17.365	41.919	9.122	17.577
memo	75	49	77	48	7.179	2.991	0.223	0.086	1.271	1.731	0.802	3.431
move	76	21	141	109	10.613	6.154	0.481	0.283	1.450	1.633	16.546	14.105
new_proc	78	3	1,054	1,356	103.244	109.141	2.946	2.490	2699.035	4665.479	22.685	10.541
pl1	80	53	855	806	99.628	51.048	6.287	4.824	31.949	56.551	12.505	8.233
pl1_abs	81	24	132	109	9.705	3.770	0.327	0.207	2.326	5.139	21.990	14.039
print	82	102	114	93	7.073	8.247	0.427	0.533	15.779	42.323	31.281	60.033
print_messages	86	1	157	00	13.302	0.000	0.407	0.000	0.399	0.000	14.262	0.000
print_request_type	89	2	52	55	3.601	4.835	0.261	0.174	0.290	0.192	50.452	50.932
print_wdir	91	5	101	65	5.180	2.199	0.322	0.514	0.176	0.155	23.034	31.531
program_interrupt	94	7	525	593	33.705	35.288	0.929	0.998	129.833	143.334	27.102	59.103
progress	95	5	2,398	2,410	261.571	125.632	19.997	10.800	79.956	106.028	18.023	3.737
qedx	96	95	1,865	2,404	76.755	90.895	3.175	3.649	699.599	1197.574	39.233	125.457
ready	97	2	83	66	3.920	3.719	0.093	0.036	5.735	7.073	4.544	5.393
ready_off	98	16	31	19	4.292	2.073	0.074	0.014	0.234	0.169	0.083	0.033
ready_on	99	15	07	03	1.129	0.560	0.023	0.001	0.031	0.011	0.022	0.010
release	100	99	87	86	5.522	5.978	0.142	0.136	0.045	0.139	12.296	17.103
rename	101	22	106	48	6.383	4.320	0.183	0.082	0.520	0.515	27.028	20.557
resource_usage	102	32	87	51	5.043	3.150	0.207	0.145	1.151	2.139	23.245	34.553
runoff	104	42	304	172	26.279	16.354	3.974	2.947	23.654	82.763	24.910	43.115
runoff_abs	105	1	37	00	1.405	0.000	0.118	0.000	0.339	0.000	4.521	0.000
safety_sw_on	107	4	108	24	8.984	2.474	0.140	0.068	0.816	0.531	33.801	34.337
send_message	108	12	208	161	11.923	7.876	0.376	0.329	14.650	30.635	43.489	44.633
set_acl	109	26	97	34	5.566	1.963	0.219	0.083	0.754	0.553	25.766	14.583
set_com_line	112	24	37	41	2.704	2.449	0.073	0.076	0.274	0.324	4.048	3.515
set_iacl_seg	114	5	92	26	5.685	2.702	0.140	0.047	0.635	0.420	27.793	5.151
set_search_rules	115	48	94	41	10.465	5.030	0.283	0.059	1.559	0.347	0.576	2.551
set_tty	116	8	97	57	9.466	6.960	0.146	0.078	0.445	0.412	23.167	24.727
start	118	33	1,429	2,108	83.933	139.345	4.709	8.392	102.591	261.134	29.216	78.647
status	119	14	155	112	11.457	10.075	0.319	0.281	1.380	1.013	16.603	11.451
terminate	120	1	112	00	7.949	0.000	0.197	0.000	1.179	0.000	43.453	0.000
terminate_refname	122	7	72	26	5.242	2.926	0.182	0.052	0.600	0.111	19.717	33.734
terminate_single_r	123	5	83	08	6.465	0.976	0.146	0.027	0.347	0.153	9.331	1.335
trace_stack	125	2	992	669	70.001	29.680	3.618	2.439	170.533	15.793	50.805	80.714
unlink	128	2	79	12	4.101	0.426	0.244	0.154	0.689	0.774	32.554	17.575
walk_subtree	129	8	440	282	26.373	15.275	3.106	4.039	20.867	23.511	25.249	14.351
where	130	53	103	76	6.996	5.061	0.195	0.114	0.665	0.775	19.249	26.037
who	131	123	89	87	5.003	4.357	0.295	0.209	0.605	0.715	35.573	122.417
not_found	132	236	64	82	4.671	13.733	0.126	0.121	0.346	0.715	15.732	34.253
TOTAL RESULTS	0	5435	214	691	15.004	55.329	0.705	4.105	30.537	255.401	22.399	72.543

APPENDIX C

Totals for All Projects

(Note - the initials s.d. stand for standard deviation.)

****
REPORT FOR ALL PROJECT.
****

Total Number of sessions: 563 Interactive: 563 Absentee: 0
Number of processes: Total: 574 Avg: 1.000
Number of quits: Total: 827 Avg: 1.438
Avg. max load units: 84.313 Avg. load units: 29.250
Average CPU secs per session: 24.499 S.D. 36.258
Average memory units per session: 440.658 S.D.: 528.346
Average page faults per session: 5,748 S.D.: 7,030
Average session length (in secs): 3227.927 S.D.: 3685.473
53 minutes 48 seconds
Number of commands used: 15591 Per session: 27.69 S.D.: 28.77
Average think time: 33.1 secs.

****

Command Name	Code Number	Count	Page Faults (mean)	(sd)	Memory Units (k) (mean)	(s.d.)	CPU Time (mean)	(s.d.)	Real Time (mean)	(s.d.)	Think Time (mean)	(s.d.)
not_monitored	1	3508	213	697	15.847	63.720	1.245	8.150	52.063	253.939	26.090	79.419
abbrev	2	100	34	32	2.144	2.272	0.068	0.069	0.079	0.079	8.351	14.731
accept_messages	3	221	87	73	9.101	6.564	0.292	0.161	1.831	5.843	0.602	5.092
add_name	4	26	118	62	7.714	4.465	0.417	0.515	3.913	8.247	20.685	24.375
add_search_rules	5	349	41	25	7.450	3.831	0.215	0.037	0.483	0.392	0.447	3.041
adjust_hit_count	6	8	80	30	6.284	4.815	0.204	0.138	0.575	0.482	12.779	5.049
answer	7	2	259	178	16.294	11.842	0.558	0.240	1.753	1.041	26.220	36.292
archive	9	185	208	142	18.934	19.037	0.478	0.282	2.246	2.243	30.834	73.508
assign_resource	10	1	65	00	4.290	0.000	0.124	0.000	0.452	0.000	2.455	0.000
basic	11	26	621	645	26.924	23.756	3.239	4.139	408.601	448.025	21.647	36.007
basic_system	12	1	147	00	7.908	0.000	0.442	0.000	10.538	0.000	6.684	0.000
bind	13	35	410	288	43.282	28.130	4.131	3.146	10.818	11.404	17.199	12.472
calc	14	6	252	261	9.406	8.401	0.272	0.249	253.595	464.852	293.657	551.932
cancel_abs_request	15	14	185	118	9.827	5.247	0.390	0.314	1.212	0.996	16.797	9.351
cancel_daemon_requ	16	8	113	49	7.321	2.271	0.215	0.106	0.769	0.804	19.957	8.797
change_error_mode	18	46	29	16	2.846	1.123	0.031	0.017	0.163	0.197	0.369	1.136
change_wdir	19	488	70	55	4.082	3.215	0.114	0.110	0.293	0.319	26.164	43.145
check_info_segs	20	31	242	112	25.612	10.315	1.973	0.405	6.733	4.429	11.356	17.194
close_file	21	13	55	36	3.489	2.631	0.087	0.048	0.431	0.566	14.671	17.833
compare_ascii	24	29	166	81	8.548	5.784	0.399	0.282	8.885	37.918	24.594	28.259
console_output	25	53	32	26	2.461	1.782	0.046	0.026	0.219	0.533	0.573	1.262
copy	26	238	95	66	7.711	7.398	0.237	0.185	1.792	4.940	46.165	50.930
create	28	6	76	48	3.857	2.844	0.166	0.096	0.585	0.427	16.043	12.550
create_dir	29	10	84	28	6.333	2.870	0.177	0.101	0.486	0.270	28.790	19.795
debug	31	5	595	242	30.742	12.320	1.167	0.598	208.256	406.313	49.312	71.857
delete	34	815	77	45	4.557	2.812	0.203	0.130	0.659	0.990	30.015	48.061
delete_acl	35	9	114	59	7.205	3.030	0.230	0.115	0.983	0.890	45.142	69.385
delete_dir	36	4	245	111	14.377	5.096	0.951	0.840	7.315	2.457	6.723	1.832
delete_force	37	1	14	00	0.210	0.000	0.073	0.000	0.061	0.000	3.579	0.000
delete_iacl_dir	38	1	124	00	6.152	0.000	0.121	0.000	2.269	0.000	62.402	0.000
delete_name	40	9	70	29	4.608	2.894	0.120	0.051	0.409	0.607	26.957	13.734
delete_search_rule	41	1	53	00	3.358	0.000	0.175	0.000	0.320	0.000	5.885	0.000
do	42	18	303	171	28.813	14.462	0.631	0.397	12.808	5.443	51.723	116.753
dprint	43	99	195	148	15.648	9.881	0.445	0.326	2.312	2.845	24.316	33.902
dump_segment	45	33	172	201	7.328	5.899	0.574	0.708	162.986	430.035	38.231	53.668

edm	46	1369	408	559	11.435	15.366	0.660	1.896	609.850	1172.414	86.256	172.983
enter_abs_request	48	15	210	87	12.686	4.434	0.372	0.187	2.223	1.487	27.608	34.732
exec_com	49	259	188	245	18.869	22.640	0.463	0.659	26.073	115.891	24.459	39.398
file_output	50	56	146	76	9.047	4.455	0.280	0.175	0.814	0.921	37.541	32.853
fs_chname	53	2	86	54	4.723	2.945	0.188	0.096	0.139	0.133	31.608	28.665
get_quota	56	78	132	108	8.969	6.113	0.278	0.252	1.285	1.863	20.920	50.725
help	57	134	191	172	10.374	8.954	0.363	0.415	49.572	75.597	29.898	44.498
how_many_users	58	82	54	65	2.235	2.827	0.091	0.102	0.348	0.469	44.492	56.403
immediate_messages	59	13	58	34	2.641	1.926	0.302	0.038	0.490	0.285	15.510	13.423
indent	60	622	74	35	4.736	3.287	0.259	0.105	0.724	0.576	23.613	38.072
initiate	61	9	67	18	4.530	2.295	0.111	0.050	0.171	0.131	8.973	5.776
io_call	62	5	104	31	5.692	2.197	0.181	0.049	0.244	0.158	30.172	23.002
line_length	63	16	55	35	2.915	2.148	0.109	0.059	0.316	0.476	24.096	49.164
link	64	84	102	78	6.230	4.791	0.188	0.175	5.701	13.301	55.017	76.491
list	65	1011	76	65	5.011	4.674	0.228	0.289	1.645	20.647	25.671	75.893
list_abs_requests	66	30	182	68	13.228	3.645	0.370	0.135	1.790	1.065	17.511	18.822
list_acl	67	50	82	49	5.115	3.636	0.200	0.128	0.815	1.072	20.758	42.891
list_daemon_reques	68	38	127	70	10.186	5.912	0.311	0.199	1.174	1.245	40.046	99.634
list_iacl_dir	69	4	85	47	4.745	2.916	0.122	0.043	0.923	1.013	14.064	14.212
list_iacl_seg	70	6	62	35	3.938	2.806	0.117	0.043	0.698	0.518	12.012	10.207
list_ref_names	71	6	58	12	3.415	1.341	0.073	0.029	0.215	0.193	27.434	10.762
list_resources	72	2	105	38	9.699	5.163	0.372	0.064	1.161	0.409	46.295	15.959
logout	73	474	91	45	6.596	3.670	0.111	0.089	0.626	0.605	65.812	130.930
mail	74	210	177	111	21.741	12.550	0.598	0.281	15.848	44.958	11.519	27.447
memo	75	49	77	48	7.179	2.991	0.223	0.086	1.271	1.701	0.802	3.431
move	76	24	134	103	10.272	5.881	0.463	0.304	1.349	1.564	20.257	18.874
new_proc	78	11	558	702	54.652	58.853	2.065	1.311	744.639	2434.962	68.619	116.641
page_trace	79	1	197	00	17.338	0.000	1.131	0.000	110.546	0.000	173.064	0.000
pl1	80	876	468	418	67.624	39.743	4.057	2.993	14.020	25.462	22.441	34.167
pl1_abs	81	28	133	106	10.250	9.184	0.330	0.227	2.628	6.036	22.664	14.194
print	82	860	102	87	4.629	4.254	0.433	0.550	75.629	146.337	40.675	61.703
print_auth_names	84	1	56	00	2.580	0.000	0.162	0.000	0.246	0.000	12.139	0.000
print_messages	86	2	88	96	6.861	9.109	0.220	0.264	0.247	0.214	28.145	19.633
print_request_type	89	2	52	55	3.601	4.835	0.261	0.174	0.290	0.199	50.452	60.909
print_search_rules	90	1	11	00	0.132	0.000	0.073	0.000	0.082	0.000	31.473	0.000
print_wdir	91	24	65	48	3.104	2.260	0.159	0.258	0.218	0.309	97.109	228.107
probe	92	97	224	175	13.974	11.658	0.394	0.355	143.560	179.922	37.606	50.599
profile	93	29	82	48	3.734	2.010	0.476	0.647	77.027	128.199	22.700	19.832
program_interrupt	94	19	325	432	18.896	25.348	0.533	0.707	117.028	155.437	17.797	37.156
progress	95	5	2,398	2,410	261.571	125.632	19.997	10.800	79.956	106.028	18.023	3.737
qedx	96	121	1,501	2,240	61.715	85.503	2.581	3.428	586.584	1084.436	43.222	115.745
ready	97	2	83	66	3.920	3.719	0.093	0.036	5.736	7.073	4.544	6.393
ready_off	98	16	31	19	4.292	2.073	0.074	0.014	0.234	0.169	0.083	0.033
ready_on	99	15	07	03	1.120	0.560	0.023	0.001	0.031	0.011	0.022	0.010
release	100	594	39	47	2.622	3.509	0.069	0.079	0.016	0.055	19.283	41.306
rename	101	48	81	45	4.764	3.735	0.144	0.090	0.749	1.714	38.883	63.019
resource_usage	102	85	61	43	3.491	2.808	0.147	0.108	0.615	1.372	28.396	43.273
runoff	104	59	264	167	23.151	15.579	3.332	2.848	52.954	137.854	26.969	38.595
runoff_abs	105	1	37	00	1.406	0.000	0.118	0.000	0.339	0.000	4.521	0.000
safety_sw_on	107	4	108	24	8.984	2.474	0.140	0.068	0.816	0.401	33.801	39.039
send_message	108	33	159	132	10.590	9.350	0.317	0.291	58.510	186.376	54.494	43.920
set_acl	109	96	71	29	4.544	3.102	0.173	0.140	0.463	0.480	64.573	212.765
set_com_line	112	24	37	41	2.704	2.449	0.073	0.076	0.274	0.324	4.048	8.515
set_iacl_seg	114	11	70	34	4.565	3.062	0.137	0.046	0.424	0.379	24.470	10.994
set_search_rules	115	48	94	41	10.465	6.030	0.283	0.059	1.558	0.847	0.576	2.651
set_tty	116	345	129	66	16.260	7.226	0.444	0.148	0.335	0.333	12.119	10.004
start	118	86	630	1,447	36.055	93.694	2.367	5.857	90.020	245.574	23.966	52.750
status	119	18	136	106	9.796	9.423	0.282	0.261	1.157	0.983	58.308	157.432
terminate	120	1	112	00	7.049	0.000	0.197	0.000	1.179	0.000	48.453	0.000
terminate_refname	122	7	79	26	5.242	2.926	0.182	0.052	0.600	0.449	19.717	33.744

terminate_single_r	123	5	83	08	6.465	0.976	0.146	0.027	0.347	0.153	9.331	1.336
trace_stack	125	2	992	669	70.001	29.680	3.618	2.439	170.533	15.793	60.805	80.714
unlink	128	86	15	22	1.076	1.045	0.073	0.040	0.139	0.195	4.638	10.557
walk_subtree	129	8	440	282	26.372	15.276	3.106	4.039	20.867	23.513	25.249	14.058
where	130	57	101	75	6.589	5.005	0.190	0.113	0.667	0.905	19.966	27.950
who	131	166	79	78	4.428	4.077	0.264	0.193	0.540	0.649	42.147	117.074
not_found	132	650	65	55	4.321	8.529	0.110	0.095	0.384	0.582	44.107	106.329
TOTAL RESULTS	0	15591	186	486	13.665	37.676	0.792	4.154	80.146	426.541	33.181	86.893