UNOS 1436

Bell Laboratories

subject: Load Characterization for SDS/UNIX
Case: 40125

date: June 27, 1979

from: V. P. Holmes
IH 3622
4G-431 x2539

B. J. Bartels Lax
IH 3622
4H-420 x5645

P. M. Lu
IH 3622
4G-432 x5868

J. L. Pappas
IH 3622
4G-431 x5861
3622-790627.01MF

ABSTRACT

Part of the performance effort for SDS/UNIX* in Department 3622 is the understanding of load for a UNIX machine in an SDS/UNIX environment and the establishment of a standard SDS/UNIX benchmark load. Knowledge of the capacity and expected performance of a UNIX machine participating in an SDS configuration is essential for engineering SDS/UNIX to meet the performance and capacity needs of a development project. In addition, performance goals are being set and refined for SDS/UNIX both at the system and subsystem level during its development so that the performance specifications of SDS/UNIX can be certified and delivered along with the product. These performance goals are being set with respect to a standard SDS/UNIX load which represents a realistic yet acceptable load for a given UNIX machine.

The following study examines four PDP 11/70's running PWB/UNIX 1.2 in existing SDS environments and characterizes the load on these machines. Included in this characterization are the metrics used to measure load, data and analysis for load measurements taken for a three day work period, and a profile of the activities or "job mix" on each machine. The job mix profiles pro-

* UNIX is a trademark of Bell Laboratories

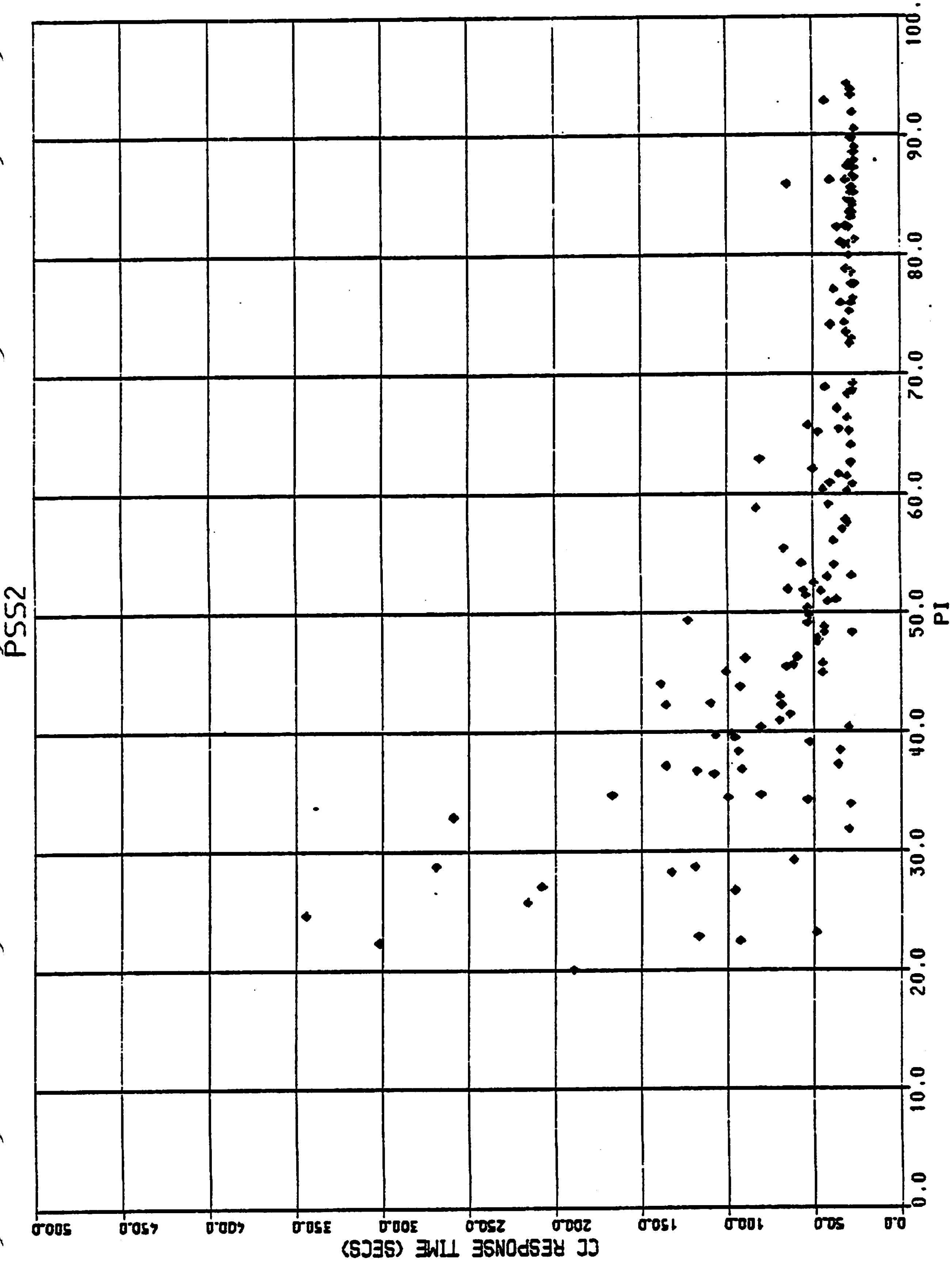
vide an hourly view of what proportion of user available CPU resource is being used by both specific commands and categories of SDS activities (e.g. SGS, networking, documentation), and frequencies of command invocations.

From the analysis of PWB/UNIX 1.2 presented in this study, a "standard SDS/UNIX benchmark load" is established to be: PI >= 40, active processes >= 15, and average process size = 23K, where PI is a Performance Index computed as a ratio of CPU (user + system) time to real or response time. The PI used in this study was found to be a reasonable estimator of system load.

The standard SDS/UNIX benchmark load is intended to represent a realistic environment for which the SDS/UNIX and its subsystems are being developed. Future work in SDS/UNIX load characterization includes a similar study for PWB/UNIX 2.0 Augmented. The standard benchmark will be refined to reflect any differences in performance characteristics.

Atts.
References
Appendixes 1-20

Copy (with att.) to
All Supervision Department 3621
All Supervision Department 3622
All Supervision Department 3627
All Members Quality Engineering and Application Support Group
T. F. Arnold
J. A. Githens
D. A. De Graaf
R. V. Epley
T. G. Hack
H. M. Jackson
J. Janik
G. W. R. Luderer
H. O. Norton
R. A. Polsley
R. A. Reed
J. M. Scanlon
M. T. Skubiak
B. A. Tague



Appendix 11.2

JOB NAME=JLP***** TSS NO=02A1 BIN=0360 NAME=PAPPAS
DATE=06/27/79 TIME=13:29:55 PAGE NO 001

CONTENTS

CONTENTS
(Continued)

Bell Laboratories

subject: Load Characterization for SDS/UNIX
Case: 40125

date: June 27, 1979

from: V. P. Holmes
IH 3622
4G-431 x2539

B. J. Bartels Lax
IH 3622
4H-420 x5645

P. M. Lu
IH 3622
4G-432 x5868

J. L. Pappas
IH 3622
4G-431 x5861
3622-790627.01MF

MEMORANDUM FOR FILE

1. INTRODUCTION

Release 1 of the common Software Development System (SDS) [1], named SDS/UNIX, is currently under development in Laboratory 362. With this release, the SDS will manage large development projects distributed across a homogeneous multi-machine environment. Despite the additional capabilities which will be available in this new SDS release, it will only be successful if it meets the needs of the users both in terms of capabilities and performance. Thus, the determination of performance goals and certification that those goals are met prior to SDS release is as important as the determination of SDS capabilities and certification thereof.

Performance goals can only be specified in terms of load. Thus, in order to set realistic performance goals for SDS/UNIX, an understanding of what factors contribute to load and what constitutes an acceptable load level is necessary. It is the purpose of this study to profile the load of existing UNIX* machines engaged in SDS activities, define load metrics, and establish an acceptable load level. This load profile exercise will illustrate the types of activities that contribute to load and the extent of these contributions to load. The definition of a load metric is necessary in order to determine at any point in time the existing load level on a machine. Finally, the establishment

* UNIX is a trademark of Bell Laboratories

of an acceptable load level or a "standard benchmark load" will be the standard against which all performance goals will be specified.

Currently there are few established UNIX load metrics and even fewer tools available to measure such metrics. Our current best indicator of load is a "gut feeling" of experienced UNIX developers. There are, however, prototype development projects for SDS/UNIX at Indian Hill, namely the DMERT project and No. 3 ESS. Both projects incorporate multiple UNIX machines in their SDS architectures and offer fundamental choices in load distribution of SDS functions (see Section 2).

Consequently, tests were run and data was collected on selected Indian Hill UNIX machines in an attempt to characterize SDS/UNIX load. From these tests data was collected in two general categories - real time data (i.e. data which sampled the performance index (PI), the number of users, the number of active processes, and the response time on a particular command multiple times per hour throughout the day) and accounting data (i.e. data collected by the system about every command which was executed). Finally, data reduction techniques were developed and used to analyze the collected data.

This memorandum covers a description of the tests executed, the characteristics of the user community on each machine analyzed, the analysis of all collected data, conclusions about viable load metrics and a standard benchmark load definition. This memorandum also profiles the existing UNIX based SDS environments in terms of the type of activities and the proportion of machine capacity used for each activity.

2. ENVIRONMENT

2.1 Machines

Release 1 of SDS/UNIX will be implemented on a homogeneous multi-machine environment. This environment will consist of two different flavors of the UNIX operating system - PWB/UNIX 2.0 augmented, and UNIX/RT - operating on PDP 11/70s.

In order to provide performance goals for the SDS/UNIX it is necessary to understand the concept of load as it applies to each of the operating systems on a PDP 11/70. Unfortunately, at the time of this study the only existing UNIX operating systems are PWB/UNIX 1.2 and MERT. Thus, in the short term in preparation for the establishment of preliminary performance goals, these available operating systems were analyzed with respect to load. As PWB/UNIX 2.0 augmented and UNIX/RT become available, load for these new environments will also be analyzed in order to determine any significant differences and the effect of these differences upon performance.

The load analysis exercise centered on UNIX C, UNIX D, IHPS1, and IHPS2. All these machines run under PWB/UNIX 1.2. The emphasis of these experiments was on these machines for the following reasons:

1. One of the first applications of SDS/UNIX is #5 ESS which is supported by an architecture that is dominated by Programming Support System (PSS) machines.

2. The user environment is strictly controlled on the LSS (Laboratory Support System) UNIX/RT machines, and the types of jobs executing on these machines are mainly related to laboratory testing. The load can be reasonably controlled by limiting the number of ports available to users.

3. One of the high-runners in CPU consumption is view configuration. This activity will be done on the PSS machines. It is, therefore, critical that the PSS machines provide the required system capacity for this activity.

The selection of these particular four UNIX machines was not arbitrary. They were selected primarily because the work being done on these machines correlated in many respects to the work which will eventually be done on SDS/UNIX PSS machines. Each machine supports a different user community that are summarized as follows.

1. UNIX C - This machine is one of the general purpose UNIX machines maintained by the computer center (Laboratory 376). The user community consists primarily of members of Laboratory 361 and Department 3623. Thus the work on this machine involves 3B and DMERT development, a C language based system.

2. UNIX D - This machine is also one of the general purpose UNIX machines maintained by the computer center (Laboratory 376). The user community consists primarily of members of Laboratory 362. This involves DMERT development in Department 3623, TUS work in Department 3627, and documentation efforts in all Indian Hill Laboratory 362 Departments. With the DMERT effort split between UNIX C and UNIX D, a large amount of network traffic is necessary between these machines.

3. IHPS1 - This machine is owned by the #3 ESS project and maintained by Department 5315. IHPS1 is the PSS machine for #3 ESS software development. However, since #3 ESS is a SWAP based project, IHPS1 interfaces to a back end TSS processor for software generation. The two interesting aspects of the use of this machine (with respect to SDS/UNIX) are its use of the Bell Labs Network (BLN) and its use of pieces of the Change Management System (CMS).

4. IHPS2 - This machine is owned by the #2 ESS project and also maintained by Department 5315. IHPS2 is the PSS machine for #2 ESS software development with an interface to a back end TSS processor through ASP network facilities for SWAP software generation. IHPS2 also uses pieces of CMS, particularly SCCS.

The UNIX C and UNIX D machines were chosen for load characterization analysis because their primary workload is C program development for DMERT. DMERT will be the first application for Release 1 of SDS/UNIX, and #5 ESS will also be a C environment. DMERT is a good example of a project that requires distributed system support, as the project is developed on a number of separate machines. IHPS1 and IHPS2 support projects that are currently using some early versions of the tools that will enhanced and supported in Release 1 SDS/UNIX. Both of these machines are configured to operate as a front end machine to a TSS machine. The load and overwrite generation is done on the TSS machines. The choice of these machines enables an effective analysis of the tradeoffs encountered between networking software generation jobs to a back end processor and local software generation.

3. STUDY

Data was gathered on the four machines described in the previous section over a three day period, June 4, 5 and 6, 1979. User communities on these machines were in preparation for conversion to PWB/UNIX 2.0 Augmented. As a result of this preparation, most users were using the Bourne Shell program, but the Mashey Shell was still available and being used. Many of the other 2.0 facilities and tools were available and are reflected in the data presented here.

One of the major difficulties encountered during the design and execution of this load characterization study was the lack of available software tools to gather and reduce load characterization data. The only system provided load measurement is the "performance index" (PI) which is measured and recorded in a file, /usr/adm/pi, every five minutes. This PI is computed by running a program (eat) which consumes system resources and measures both the real (elapsed) time and CPU (user + system) time for the execution of the program. The ratio of CPU time to real time is computed for the execution of the program and multiplied by 100 to get a numeric value between 0 and 100 representing PI. In addition to each PI entry, the trivial response time is measured and recorded along with the PI.

For this study, tools were developed to both capture and reduce load measurement data. A shell procedure (PDAEMON) was written to run continuously and take real-time measurements at a specified interval of time. At a regular interval of activation, it would measure the number of active users and active processes and measure the response time for particular commands. The data from this procedure was logged in a file for subsequent analysis. In addition the UNIX accounting files [2] were processed by a collection of programs to capture the hourly "job mix" for each machine. This information includes hourly frequency of invocation and CPU usage.

3.1 PDAEMON Data Collection

PDAEMON is a Bourne Shell procedure that was run continuously on each machine. Every 15 minutes it would gather the following information:

- date and time,
- number of active users,
- number of available processes,
- number of active processes,
- time for a file transfer,
- time for a compile.

The number of available processes was computed by measuring the size of the process table. The active processes were computed by eliminating the non-active processes (vacant entries) and eliminating the user shells. The file transfer was a .p of a 41K file. The compile used the available C compiler to compile a 20K source with a 1.5K header file.

A sample of the PDAEMON data can be found in Appendix 1.

3.2 Job Mix Data Collection and Reduction

Though most users have a feel for some of the information gathered by PDAEMON, it is the job mix of the machines involved that can provide new insights into load characterization for both SDS designers and users. What was uncovered through this study was the percentage of CPU usage that could be attributed to both individual commands and categories of commands for a given hour, and the hourly frequencies of invocation for all commands. It was felt that the hourly sampling granularity best reflected the job mix activity patterns both during prime shift and non-prime shift.

The job mix data was generated by processing the UNIX accounting files [2]. Upon termination of every process, the operating system writes a record of information about the invocation and execution of that process in a file, /usr/adm/pacct. See acct (V) [3] for the exact structure of the accounting record. The file can subsequently be read and formatted by the acctcom command [3] to produce a chronological listing of all terminating processes on a machine.

In order to acquire the hourly job mix data (see Appendix 2), the acctcom command was modified to produce two different views of the mix. One view is all the processes initiated during a given hour. If a process is invoked during the hour and does not terminate for several hours, it is recorded as the activity for the hour in which it was initiated only.

The second hourly view was determined by capturing those processes that existed during the hour. The CPU usage for a process attributed to the hour was that proportion of real time for the process that occurred during the hour. For example, if a process ran for one hour of real time from 8:30 to 9:30, the command was logged in both hours (8 and 9) and 50% of the CPU time was attributed to each. A closer approximation of CPU usage for a given hour was obtained and the frequency entry in these hourly reports (see Appendix 2) represents the number of processes that were either initiated or existed during an hour.

Both job mixes were collected for the hours from 5 a.m. to 7 p.m. for the four machines over the three day period of PDAEMON data collection. To reduce the hourly job mix files, the files were summarized with acctcms [3] and subsequently categorized into summary reports that reflect major subdivisions of machine usage. To accomplish this categorization, a data base of UNIX commands was constructed and used to look up commands appearing in the summary reports. The data base contains every command invoked on the four machines and each is a subelement in one of the following ten categories:

- CMS includes CMS commands and all other change and source management commands such as SCCS and IMRS commands,

- DB includes all data base and data base-like (diff, grep, etc.) commands,

- DOC includes nroff, use of editors, and other support document tools,

- IO includes all input and output commands, e.g. cp, cat, opr

- MISC includes known commands but ones that don't fit into the other available categories, e.g. eat, sleep

- NW includes networking commands such as bln, con and usend operations,

- SGS includes all software generation tools and generation support tools; compilers, loaders, debuggers, etc.,

- SYS includes system overhead and maintenance, e.g. accton, clock, logtty
- UNI is a category for all unidentified commands,
- USR includes all user overhead commands such as pwd, find, ls, cd, etc.

The commands were categorized with the help of the users of the machines sampled and there may be some discrepancy as to the unique categorization of some commands. A sample categorized summary report can be found in Appendix 3, and the data base can be found in Appendix 4.

4. ANALYSIS

With any data collection and statistical analysis thereof, discrepancies in or assumptions about the data will be found and made. This section documents these caveats in an attempt to avoid an incorrect interpretation of the data. The PDAEMON data, the job mix data, and finally the effect of the data gathering itself on the monitored systems will be discussed.

The PDAEMON data was used as an indicator of the response time of specific commands as a function of load. These commands were repeated every 15 minutes, and various system parameters were collected. At the time this data was collected, a method was not available to determine the load immediately prior to the execution of the command. The load metric PI was extracted from the operating system's monitor, which collected this value at five minute intervals. The PI values used in these analyses are therefore within five minutes of accuracy. All future load characterization efforts (see section 5.3) will use a mechanism to compute PI in real time in order to accurately correlate load with performance on a per command basis.

The job mix data analysis is very dependent on the accuracy of the command categorization. Six hundred and twelve commands were encountered and were classified in ten categories. There were 182 commands in the UNI category that were unidentifiable either because they are private user written commands, undocumented, or ambiguous (e.g. a.out although frequently executed could be the execution of <u>different</u> C programs for the same user). Although the UNI category is 30% of the commands, this category typically represents less than 10% of the CPU used per hour. Appendix 5 shows the analysis of the UNI category with an explanation of the discrepancies.

A second caveat with the job mix data concerns the method used by the operating system to collect the command information. This data is collected and saved in the pacct file when the job <u>terminates</u>. Thus, if a process is continually active, it will not be collected, and thus never appear in any accounting reports. Consequently, the job mix data does not include data about continually active processes.

Finally, an analysis has been done to determine the effect of the performance monitoring tools on the performance and load of the system being monitored. The PDAEMON data was collected by exercising a command procedure four times hourly for fourteen hours on four machines. This PDAEMON procedure did create an extra load on the system although analysis has shown that this extra load was minimal. Appendix 6 lists the commands executed by the PDAEMON, their frequency per PDAEMON invocation, the CPU usage per PDAEMON invocation, and their total CPU usage per hour. These numbers were obtained by averaging across all four machines. Thus the PDAEMON increased the load by the issuance of 60 commands per hour at a cost of approximately 2.28 CPU minutes per hour or less than 10% of the CPU during prime time hours.

The job mix data was collected via existing system accounting procedures which while they contribute to system load are constant across all machines and are not additional overhead introduced for the purpose of performance monitoring. The job mix data reduction procedures were primarily run at night and thus had no effect on the hours being measured. The exception to this is one hour on UNIX D (1 p.m. on June 6, 1979) when the job mix reduction tools used 10% of the total CPU used that hour. The other percentages shown in Appendix 7 for UNIX D reflect not the use of the data reduction tools but rather the development effort required to build and debug those tools.

4.1 PDAEMON Analysis

There are many system parameters which are thought to reflect a system load. PDAEMON was to monitor an assortment of these parameters to discover which showed the most promise as load metrics. Absolute claims are difficult to make on data gathered over three days. These results are intended to show trends, consistencies across machines and provide ball park numbers for load characteristics.

4.1.1 Performance Index (PI) A feeling for load can be a very subjective evaluation. For instance, an experienced programmer can usually log onto a machine and immediately form an opinion from the terminal response. Since the response time is most visible to the programmer it was felt that a parameter based on this factor, such as PI, would show the most promise in quantitatively measuring this factor.

PI was plotted versus time for all four machines over the three day period and is included in Appendix 8. The resulting plotted points are fairly scattered but some patterns do emerge. All graphs show that PI usually has two minimal periods of the day and three peaks or periods of maximum. The results show a definite double camel-hump like pattern which is well known to be associated with human activity. The data for IHPS1 do not show this pattern quite as strikingly as do the rest but this may be due to the large amount of constant network activity placed on this machine.

PI typically dips to a low in the 20's from around 8 to 10 in the morning. The values increase from 11 am to 1 pm, reflecting the lessening activity over lunch hour. Then PI falls again to the 20's from around 1 p.m. to 5 or 6 p.m. For non-prime time hours the PI usually stabilizes at its maximum for that machine. UNIX C, UNIX D, and IHPS2 usually reach a maximum of around 90 to 95 during this period. The maximum PI for IHPS1 is consistently lower at a value of around 65. Again this discrepancy is likely due to a consistent network overhead not on the other machines.

Thus, PI can be used as a reasonable indicator of system load. This is further shown by a comparison of PI with the compile time data gathered from PDAEMON. That is, when the PI decreased, compile times increased and, conversely.

4.1.2 Active Users versus Active Processes

Active users and active processes were two measurements taken by PDAEMON. A correlation between number of processes and number of users would be useful for obtaining a gross number of machines that a project needs to acquire in order to meet its capacity needs.

To determine the exact relationship, the PDAEMON measurements, taken every 15 minutes, were plotted and a linear curve was computed for the points on each machine. The resulting graphs can be found in Appendix 9 and the following equations and coefficients of determination ($0 <= D <= 1$) were derived (let u = active users, p = active processes). These coefficients of determination indicate how well the points fit the line where a low coefficient implies a poor fit.

UNIX C: p = .26*u + 13.94, D = .18
UNIX D: p = .59*u + 11.32, D = .41
IHPS1: p = .95*u + 12.68, D = .48
IHPS2: p = .75*u + 10.89, D = .59

The analysis shows that for a given number of processes on a machine, one can only estimate a gross range in the number of users that could be logged on the system. At times there can be 20 active processes with as few as zero and as many as 27 users on the system. The general purpose machines often have more users than active processes, i.e. users logged on with no active work, whereas IHPS1 has a threshold of 13 active processes. One can only project the number of users that can be supported by the machine given the number of allowable active processes by assuming that all users are active and are invoking at least one process to complete a system transaction.

Active processes do impact the load of a UNIX machine (see Section 4.1.3), and based on the fact that 15 processes is a reasonable number for ensuring a PI of over 40, one can estimate that the number of active users that machine can support will be around 15. A better correlation between number of processes and number of users can be obtained if the user activities or scenarios were more explicitly determined. These scenarios have not been defined in this study for the standard SDS/UNIX benchmark load; therefore the number of processes a given user would spawn for completing one job was not determined.

The number of users for a given number of active processes on a machine can be better estimated if the anomaly cases can be identified and treated separately from the rest of the data for the purpose of obtaining a better correlation.

4.1.3 <u>PI versus Active Processes</u> Appendix 10 contains a plot of the number of active processes versus PI for both IHPS1 and IHPS2. There is no apparent pattern to the IHPS2 data. IHPS1 does show a trend of more active processes as the PI decreases. Another way of interpreting these graphs is to follow the value of a given PI up the vertical axis noting the range of values of number of active processes. Thus, one may determine a range of number of active processes for a given PI. Also the minimum value in this range is the least number of active processes known for this PI. Obviously, the range of number of active processes is due to the variation in job mix. As of this time, the data is not complete enough to specify a SDS/UNIX job mix to associate with a value of number of active processes. See Section 5.3 for future work in this area.

4.1.4 <u>PI versus Response Time</u> Plots for IHPS1 and IHPS2 are shown in Appendix 11 indicating the values for compile time versus the approximate PI at that time. As mentioned before, the PI is approximate since the actual PI value could not be synchronized with the timing of the compile. One observation to note here is the larger variance in compile times at lower values of PI. The data does show that larger compile times can be expected for lower PI values. For example, compile times can be expected to be twice as long or longer for values of PI below 40, as opposed to above 40. From this analysis, a PI=40 is significant and thus contributes to the definition of a standard benchmark load (see section 5.1).

4.2 <u>Job Mix Analysis</u>

4.2.1 <u>High CPU Runners</u> The categorization of the job mix data makes possible an analysis of what types of work are being done on each machine and to what extent that type of work contributes to the load on the machine. Consequently, every hour on every machine was analyzed and the top three categories in terms of percentage of CPU time used that hour was documented (see Appendix 12). In order to further refine this data the following reduction method was used:

1. each category which placed first in an hour received three points for that category for that machine

2. each category which placed second in an hour received two points for that category for that machine

3. each category which placed third in an hour received one point for that category for that machine

4. all the points were tabulated and are presented in Appendix 13.

The following are the top four categories derived by this method for each machine in high to low order:

1. UNIX C - USR, SGS, DOC, NW

2. UNIX D - SGS, USR, DOC, NW

3. IHPS1 - USR, NW, DB, MISC

4. IHPS2 - USR, SGS, DOC, MISC

Analysis of these high runner categories reveals that USR, SGS, DOC, and NW appear as high runners in at least three of the four machines. In particular, the USR category is the highest runner in three machines and second in the fourth machine. This category represents the "typical" work done by any and all users in the course of doing their job e.g. who, ls, cd, etc.

4.2.2 <u>Proportion of CPU Used for Each Category</u> The job mix data was also analyzed to determine the %CPU usage and average command invocations per category on every machine. This analysis included only the prime time hours, i.e. 8 a.m. to 5 p.m.

	UNIX C	UNIX D	IHPS1	IHPS2
SGS	20	29	5	17
DOC	13	15	6	12
NW	13	10	21	7
DB	4	3	13	6
IO	5	4	4	6
USR	23	19	28	20
SYS	3	5	2	2
MISC	5	6	7	6
UNI	6	4	4	10
CMS	7	5	11	11

Figure 1
Average %CPU for Categories

	UNIX C	UNIX D	IHPS1	IHPS2
SGS	70	124	52	79
DOC	12	16	13	11
NW	79	76	91	43
DB	60	34	160	44
IO	71	69	231	76
USR	970	782	1028	753
SYS	104	102	120	140
MISC	49	40	41	39
UNI	180	17	3	24
CMS	40	21	450	59

Figure 2
Average Command Invocations

The NW, DOC, and SGS categories were individually graphed. These graphs are discussed below.

1. Appendix 14 -

   %CPU time used for networking vs. time
   # of networking commands executed per hour vs. time

2. Appendix 15 -

   %CPU time used for documentation vs. time
   # of documentation commands executed per hour vs. time

   The documentation category included all editor work. For these plots the editor percentage of CPU time and # of commands were excluded. Thus these plots are exclusively nroff and related commands.

3. Appendix 16 -

%CPU time used for software generation activities vs. time
# of SGS commands executed per hour vs. time

The graphs for the NW, DOC, and SGS categories are in terms of percentage of CPU time used. This is somewhat deceiving because a machine can be very lightly loaded and one category could be using a very high percentage of a very small amount of CPU minutes. A graph in Appendix 17 plots CPU minutes used per hour versus time.

4.2.3 Analysis of Specific Categories The networking, documentation, SGS, MISC, and data base categories were chosen for further analysis because they were the high runner categories from the previous sections in terms of the percentage of CPU used. The CMS category was also chosen for further analysis because of its importance to SDS/UNIX.

4.2.3.1 Networking

All four machines used a significant amount of CPU per hour on networking. IHPS1 was the only one that was supported by BLN services. The high-runner in the category of networking in terms of frequency of execution was "hasp". The average amount of CPU consumed by "hasp" over the four machines was around one CPU minute per hour, with an average frequency of 15 to 20 invocations per hour.

The statistics collected on the three days for IHPS1 indicated the following results :

1. BLN consumed an average of 4 to 6 CPU minutes per hour, with a rate of no less than 2 CPU minutes per hour even when the system was not processing any network requests.

2. The frequency of invocation averaged around 8 to 10 BLN user requests per hour. This resulted in an average of 24 sec. CPU time per job.

3. There were numerous restarts over the day for each day that was monitored. The average number of restarts was around 6 times a day. (This is calculated from the number of "BLN.PS1" invocations. Restarts are usually due to BLN network system errors or failures.)

In comparison with other networking activities in the system (such as "hasp", "con", "tss", "rje", "uucp", etc.), BLN seemed to consume much more system CPU time than all the other network jobs combined. The amount of CPU consumed by these other network commands averaged 1.5 to 2 minutes of CPU per hour. The frequency of invocation for all other jobs was higher, with an average of 25 to 30 user requests per hour over prime time. This averages to about 3 sec CPU time per job. These averages are gross estimates of the average amount of CPU time used for each network request job, and is useful for projecting the system's resource used given a specific network load. Appendix 18 provides a graphical illustration of the amount of CPU consumed by BLN versus the amount consumed by all other network processes with the frequency of user requests denoted in brackets on the curves.

A comparison of the PI value for IHPS1 and other machines (as shown in Appendix 8), one observes that all other machines achieve a PI of 95 when the system is fairly idle, whereas IHPS1 never exceeds 65. This implies that jobs on IHPS1 could never obtain a faster turnaround time than 1.5 times the amount of system time required to process a given job. By examining the job mix data, it can be found that the job that caused the deterioration of the PI was the BLN process.

Interesting comparisons were made by observing the results summarized in the graphs shown in Appendix 2 .

1. Networking on UNIX C, UNIX D, and IHPS2 consumes about 12% of the total CPU used for the hour on each machine. Networking on IHPS1 consumes about 25% of the system's available resource for the hour.

2. IHPS2 and IHPS1 both depend on a back end machine for software generation. The amount of networking in terms of number of user invocations compared to other categories on the same machine ranks higher than on the UNIX C and UNIX D machines.

   (NOTE: The number of commands indicated on the figures reflect the number of active network processes, and do not correspond to user invoked commands. For example, one BLN user request requires up to 4 BLN processes to be active in order to process and complete 1 user request.)

The high CPU consumption for networking on IHPS1 could be the fact that BLN consumes a blanket overhead of 2 minutes CPU per hour when no network requests are outstanding. The statistics show that other network jobs do not consume any significant system resource when they are idle.

In the graphs (Appendix 17) that show the total amount of CPU used for the hour for each machine, we observe that the lowest threshold for CPU minutes used in an hour does not drop below 8 minutes for IHPS1, whereas other machines idle at around 4 minutes. The average amount of CPU used in an hour for each of the machine is :

1. 27 minutes for IHPS1,

2. 25 minutes for IHPS2,

3. 30 minutes for UNIX C,

4. 30 minutes for UNIX D.

In analyzing the data on the proportion of CPU used in each category, it can be estimated that networking (after subtracting documentation, miscellaneous, and unidentified) consumes about:

1. 9.8% on UNIX C,

2. 7.5% on UNIX D,

3. 5% on IHPS2,

4. 38.4% on IHPS1.

4.2.3.2 Documentation Documentation is another category which ranks very high on UNIX C, UNIX D, and IHPS2. Comparison of the documentation graphs in Appendix 15 reveals that UNIX D has, in general, the largest percentage of time devoted to documentation. High peaks occur frequently but also during periods of high total CPU utilization (see Appendix 17). In terms of comparison the high percentage CPU peaks on UNIX C and IHPS2 are not as consistently pronounced and frequently occur at times when total CPU utilization is not great. IHPS1, in complete contrast, has a fairly small amount of documentation work.

A possible method of decreasing the load on a UNIX machine is to eliminate the documentation efforts. From Figure 1 it can be seen that this would decrease the load on the machines by approximately 12-15% excluding IHPS1 where only a small amount of documentation is done. Although 15% represents a significant amount of reclaimed CPU on a heavily loaded machine such as UNIX D, documentation is still an important aspect of development work and can therefore not be completely eliminated.

There are three possible ways to handle documentation:

1. completely offload it to a separate machine

2. continue the documentation editing on a local machine and network the changed file to another machine to be processed

3. queue all documentation jobs for non-prime shift execution.

The first method is possibly a good short term solution although in the longer term it contradicts SDS efforts to make documentation an integral part of development. The second method offloads the nroff and related high CPU usage jobs to another machine although a substantial price will be paid for networking. The increased network usage will probably offset all CPU gains made by offloading documentation. Finally, the third alternative keeps documentation and development on the same machine without penalizing prime shift hours. This methodology should be considered particularly for large, very expensive documentation jobs.

4.2.3.3 CMS Although not one of the high runner categories, the CMS category must be analyzed because it is a part of Release 1. From Appendix 13, CMS ranks fairly high for both IHPS1 and IHPS2. Although #2 ESS and #3 ESS are not using a "full-blown" CMS, a trend can be established in that as more CMS capabilities are used the relative importance of the CMS category will increase. This is particularly critical on UNIX C and UNIX D where, considering the existing load on these machines, the increased capacity needed for the use of CMS will not be available.

4.2.3.4 Software Generation The SGS category exhibits very high usage on UNIX C, UNIX D, and IHPS2. This is to be expected on UNIX C and UNIX D because these machines carry the DMERT C development effort. However, IHPS2 is not a C but rather a SWAP development machine. Closer inspection of the SGS work on IHPS2 revealed that primarily microprocessor (MAC 8) tools were being used. It was finally determined that in the time frame being monitored, importation and verification of these MAC 8 tools had been done on IHPS2 in preparation for releasing them to other development machines. Thus, the SGS effort on IHPS2 is atypical. A more typical SGS day on IHPS2 is probably more comparable to the data gathered for IHPS1.

4.2.3.5 Miscellaneous Category It was somewhat unexpected that the MISC category ranks as the fourth high runner on both IHPS1 and IHPS2. The MISC category is significantly lower percentage on both UNIX C and UNIX D. The primary contributor to the CPU time in MISC is the eat command. This command is initiated regularly 36 times per hour on every machine for the purpose of computing PI. The 36 invocations require approximately 1.5 CPU minutes. This is a very small amount of CPU time and the implication can be made that when such a small amount of CPU time constitutes such a large percentage of the total CPU time per hour then the machine must be fairly lightly loaded. A closer inspection reveals that the MISC category becomes a high runner particularly during the non-prime shift hours monitored i.e. 5 a.m.-8 a.m. and 5 p.m.-7 p.m. Thus, the work load on IHPS1 and IHPS2 is concentrated during prime time hours with light usage at all other times. In contrast, UNIX C and UNIX D have more sustained higher usage.

4.2.3.6 Data Base Category The data base category ranks high for IHPS1 principally because of fairly infrequent but very costly use of the sort command and very frequent use of the inexpensive grep command. Statistics about these two commands can be found in Appendix 19.

5. RECOMMENDATIONS

5.1 Standard Benchmark Load

In order to set performance goals for SDS/UNIX, it is essential to provide a standard load or context in which these goals are to be met. This load must be reproducible and represent a realistic SDS activity level. From the load metrics and data explored in this study, the following represents the initial SDS/UNIX standard benchmark load that defines a reasonable range of activity and a load that is commensurate with the load measured on existing SDS machines:

PI >= 40,

number of active processes >= 15,

average size process = 23K.

This load definition is bound between the number of active processes and the PI. It was derived by considering a reasonable response time for the test compile executed by PDAEMON. In Appendix 11, the response time gets progressively worse when PI drops below 40; above 40, variations in PI have little effect on response time. By first establishing a lower bound for load, a PI of 40, the correlation between PI and active processes was examined (Appendix 10) to determine a range of active processes that typically produces a PI of 40 or greater. This range was found to be 15-30 processes; however, since the PI boundary will eventually restrict the number of active processes, there is no need for an explicit upper limit. Of the four machines examined in this study, it was determined that the average size process is 23K (during prime shift, 8 a.m. to 5 p.m.). To ensure a realistic load, the standard SDS benchmark must also consist of processes that have an average size of 23K. If actual SDS processes are larger, the threshold of active processes may be lower. This restriction will additionally narrow the range of the standard load.

5.2 Application to the Simulator

This load characterization effort has been undertaken in order to establish performance goals because performance goals can only be specified in terms of load. Another aspect of performance goal establishment is the determination of the feasibility of those goals. For this endeavor a simulator model for the SDS is being designed and implemented [4]. This model requires a great deal of input about the design of the SDS system, the frequency of command execution, the resources consumed by those commands, and non-SDS related activities on the machine being modeled. In particular, two types of data have been extracted from this load characterization effort strictly because they are necessary input to the simulator. They are 1) the amount of non-SDS related work which consumes system resources and 2) a suitable time period to be modeled.

The first type of data, the amount of non-SDS related work, includes in particular the USR, MISC, and SYS categories from the job mix categorization effort. An analysis was done on these three categories to determine how much "background" noise needs to be a part of the simulator model. As stated previously, the USR category was one of the high runners on all four of the categorized machines. During peak hours (8 a.m. - 5 p.m.) the USR category (averaged across all four machines) used approximately 22.5% of the CPU which equates to approximately 6.6 CPU minutes per hour.

The MISC and SYS categories were, on the other hand, considerably less active and very consistent across all hours on all machines. The SYS and MISC categories used approximately 3% and 6% respectively of the CPU or .34 and 1.39 CPU minutes per hour respectively.

An analysis was done of how many commands actually initiate and terminate within an hour boundary. These results are tabulated in Appendix 20 with the basic result that approximately 97% of the commands fall within this boundary. Thus, an SDS simulator model could be run for an hour of simulated time with confidence that the workload being generated by that model (i.e. SDS work and background noise) realistically constitutes the entire workload for that hour.

5.3 Future Work

The current study characterized load on PWB/UNIX 1.2 machines doing the programmer support (PSS) type activities of SDS. Now that the machines at Indian Hill are converting to PWB/UNIX 2.0 Augmented, future work on load characterization includes a study, similar to the one presented here, for the same machines under the new operating system.

In addition, load characterization is to be extended to the laboratory support machines running MERT and UNIX/RT. Though the load on these machines is more controllable through administration, an understanding of lab support machine capacity through load characterization can assist in this administration for making best use of the laboratory resource and provide performance feedback to the LSS (Laboratory Support System) developers.

It is also difficult at this time to specify the exact number of users to be supported by the standard benchmark load without specifying the user's work scenarios. The number of users should be specified in terms of a job mix and user scenarios. What is needed is a mechanism for concurrently measuring load and job mix.

6. CONCLUSIONS

Real time and accounting data from four selected Indian Hill UNIX machines was collected over a three day period and subsequently analyzed. This analysis resulted in a determination of a standard benchmark load definition (i.e. PI >= 40 and the number of active processes >= 15) and a load characterization for each machine. In essence, these load profiles reflect the usage of those machines (i.e. UNIX C - USR,SGS,DOC, and NW; UNIX D - SGS,USR,DOC, and NW; IHPS1 - USR,NW,DB, and MISC; and IHPS2 - USR,SGS,DOC, and MISC), and will be used to characterize a Release 1 SDS/UNIX machine during performance goal establishment.

7. ACKNOWLEDGEMENTS

We wish to thank Roger Polsley for his assistance in helping us modify and understand the UNIX accounting programs.

V. P. Holmes

B. J. Bartels Lax

P. M. Lu

IH-3622-VPH/BJBL/PML/JLP-mm J. L. Pappas

Atts.
References
Appendixes 1-20

References

[1] B. J. Bartels Lax, R. V. Epley, T. G. Hack, C. J. Harkness, L. W. Kirchhoff, P. M. Lu, D. M. Norton, "Common SDS Architecture", 2522-790507, May 7, 1979.

[2] J. L. Pappas, "After-the-Fact Command Monitor on UNIX", Programmer's Notes, May 9, 1979.

[3] "PWB/UNIX User's Manual", edition 1.1, May 1977.

[4] V. P. Holmes, "Proposal for SDS/UNIX Simulation Model", to be published.

Sample PDAEMON data

```
Mon Jun  4 07:52:21 CDT 1979 ——— date & time
num users
      4 ——— active users
num processes
     71 scratch ——— processes
active processes
1450
214
      6 scratch ——— active processes
1
      6      39 scratch
file transfer

real         4.0
user         0.0
sys          0.7
compile

real        38.0
user        16.9
sys          4.9
Mon Jun  4 08:08:29 CDT 1979
num users
      5
num processes
     77 scratch
active processes
1700
384
     11 scratch
1
     11      63 scratch
file transfer

real         5.0
user         0.0
sys          0.8
compile

real        38.0
user        17.1
sys          5.2
Mon Jun  4 08:24:45 CDT 1979
num users
      8
num processes
     81 scratch
active processes
1757
392
     12 scratch
1
     12      69 scratch
file transfer

real        25.0
user         0.0
sys          1.0
compile

real      6:17.0
user        18.1
sys          8.7
Mon Ju  4 08:40:51 CDT 1
num us
     1
```

Appendix 1.1

```
num processes
     72 scratch
active processes
1472
257
      9 scratch
|
      9      44 scratch
file transfer

real        9.0
user        0.0
sys         0.9
compile

real     2:17.0
user       17.8
sys         6.6
Mon Jun  4 09:07:05 CDT 1979
num users
     22
num processes
    107 scratch
active processes
2786
961
     32 scratch
|
     32     163 scratch
file transfer

real       46.0
user        0.0
sys         1.0
compile

real     5:58.0
user       18.4
sys         9.9
Mon Jun  4 09:32:27 CDT 1979
num users
     24
num processes
     96 scratch
active processes
2555
519
     19 scratch
|
     19      88 scratch
file transfer

real        9.0
user        0.0
sys         1.1
compile

real     3:00.0
user       17.5
sys         8.2
Mon Jun  4 09:54:37 CDT 1979
num users
     24
num processes
     93 scratch
active processes
```

Appendix 1.2

PSS1

Summary of PDAEMON DATA

	usr	ass. proc.	active proc	file xfr.	cc_rt	cc_pt
Mon Jun 4 07:49:01	3	52	11	4.0	28.0	21.70
Mon Jun 4 08:04:48	4	60	16	7.0	39.0	21.80
Mon Jun 4 08:20:57	5	58	16	4.0	30.0	21.60
Mon Jun 4 08:36:49	9	55	14	8.0	38.0	21.80
Mon Jun 4 08:52:47	10	56	15	4.0	27.0	21.50
Mon Jun 4 09:08:32	13	53	12	5.0	1:21.0	23.00
Mon Jun 4 09:25:19	0	62	16	55.0	5:39.0	49.80
Mon Jun 4 09:50:05	15	59	19	7.0	3:17.0	24.10
Mon Jun 4 10:09:37	16	65	23	10.0	54.0	23.40
Mon Jun 4 10:26:47	16	63	19	6.0	1:20.0	23.40
Mon Jun 4 10:44:06	14	62	22	11.0	1:20.0	25.00
Mon Jun 4 11:01:08	14	69	22	16.0	3:53.0	25.20
Mon Jun 4 11:21:27	10	54	13	4.0	29.0	21.60
Mon Jun 4 11:37:16	9	54	13	4.0	34.0	21.80
Mon Jun 4 11:53:10	9	57	14	7.0	1:38.0	25.80
Mon Jun 4 12:10:30	13	60	19	4.0	37.0	20.80
Mon Jun 4 12:26:37	13	59	18	5.0	47.0	24.40
Mon Jun 4 12:42:51	12	58	16	5.0	42.0	23.50
Mon Jun 4 12:59:13	0	60	16	48.0	4:22.0	53.60
Mon Jun 4 13:22:54	16	61	19	4.0	42.0	22.10
Mon Jun 4 13:38:57	19	63	21	4.0	31.0	21.50
Mon Jun 4 13:54:56	20	82	29	6.0	2:25.0	24.80
Mon Jun 4 14:14:55	17	87	32	10.0	4:29.0	25.20
Mon Jun 4 14:37:53	15	65	20	23.0	3:02.0	23.70
Mon Jun 4 14:59:18	0	66	27	9.0	47.0	6.50
Mon Jun 4 15:17:50	18	82	34	13.0	2:18.0	25.10
Mon Jun 4 15:36:18	21	83	36	8.0	57.0	24.40
Mon Jun 4 15:53:09	21	56	15	5.0	37.0	22.10
Mon Jun 4 16:09:13	20	60	18	5.0	1:34.0	24.20
Mon Jun 4 16:26:22	18	69	24	17.0	1:28.0	23.00
Mon Jun 4 16:43:58	14	58	17	5.0	1:11.0	23.70
Mon Jun 4 17:00:48	9	65	18	8.0	1:13.0	24.30
Mon Jun 4 17:17:59	5	53	12	5.0	28.0	21.80
Mon Jun 4 17:33:46	4	49	8	4.0	29.0	21.30
Mon Jun 4 17:49:33	5	54	13	5.0	1:02.0	23.10
Mon Jun 4 18:05:58	4	58	15	4.0	41.0	21.80
Mon Jun 4 18:22:01	3	53	12	5.0	27.0	21.60
Mon Jun 4 18:37:47	1	53	13	4.0	27.0	21.40
Mon Jun 4 18:53:33	1	53	12	4.0	27.0	21.70
Mon Jun 4 19:09:19	1	53	12	4.0	33.0	21.60
Mon Jun 4 19:25:12	1	57	15	4.0	27.0	21.40
Mon Jun 4 19:41:05	1	53	12	4.0	27.0	21.60
Mon Jun 4 19:56:51	2	54	13	4.0	26.0	21.40
Mon Jun 4 20:12:37	2	54	13	4.0	27.0	21.60
Mon Jun 4 20:28:22	2	54	13	4.0	27.0	21.50
Mon Jun 4 20:44:07	2	54	13	4.0	27.0	21.40
Mon Jun 4 20:59:53	2	57	13	7.0	32.0	22.00
Mon Jun 4 21:16:02	4	55	14	4.0	28.0	21.50
Mon Jun 4 21:31:48	4	59	16	5.0	35.0	21.70
Mon Jun 4 21:47:46	3	54	13	4.0	32.0	21.60
Mon Jun 4 22:03:37	2	64	16	4.0	47.0	22.50
Mon Jun 4 22:19:54	2	58	13	6.0	32.0	21.90
Mon Jun 4 22:35:56	2	53	12	4.0	26.0	21.20
Mon Jun 4 22:51:40	2	53	12	4.0	26.0	21.30
Mon Jun 4 23:07:25	2	53	12	4.0	27.0	21.50
Mon Jun 4 23:23:11	2	57	13	6.0	44.0	23.10
Mon Jun 4 23:39:28	2	53	12	6.0	52.0	22.40
Mon Ju[illegible] 23:55:43	[illegible]	61	18	8.0	1:50.0	25.50
Tue Ju[illegible] 00:13:27	[illegible]	68	20	13.0	1:26.0	24.40

Appendix 1.3

Sample Hourly Job Mix DATA

ihuxd
jbmx0606(1) 10am — processes initiated for 10 am.

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR
TOTALS	1353	743.49	35.07	3455.60	21.20	0.03	0.01
nroff	12	173.30	6.08	50.70	28.50	0.51	0.12
3bld	10	103.14	1.95	14.00	52.88	0.20	0.14
nspeak	4	37.07	0.46	12.13	79.82	0.12	0.04
ex	4	33.11	0.60	39.68	55.46	0.15	0.02
aspd	28	27.83	1.57	37.18	17.71	0.06	0.04
sgen	2	24.13	0.41	9.92	58.68	0.21	0.04
haspmain	6	20.91	3.35	415.07	6.25	0.56	0.01
haspqer	14	20.56	0.96	14.43	21.35	0.07	0.07
bsh	121	19.02	1.80	1643.80	10.56	0.01	0.00
make	3	18.94	0.51	26.93	36.79	0.17	0.02
du	1	18.30	1.88	9.73	9.71	1.88	0.19
cpp	6	17.19	0.48	2.92	35.76	0.08	0.16
pr_3bldp	2	14.72	0.26	16.92	57.40	0.13	0.02
tc1	5	11.19	0.29	1.72	38.84	0.06	0.17
c1	3	11.17	0.36	2.33	31.35	0.12	0.15
eat	36	9.91	1.46	4.23	6.81	0.04	0.34
sh	200	9.87	0.88	168.55	11.26	0.00	0.01
tc0	9	9.00	0.22	2.75	40.14	0.02	0.08
ld	6	8.71	0.35	4.32	24.94	0.06	0.08
tc2	1	8.66	0.26	1.23	33.71	0.26	0.21
c0	3	8.61	0.28	1.92	31.02	0.09	0.14
ftp	4	7.76	0.75	8.25	10.34	0.19	0.09
ls	50	7.22	0.50	5.92	14.58	0.01	0.08
e	36	6.80	0.71	192.58	9.52	0.02	0.00
uucico	9	6.69	0.14	12.30	49.26	0.02	0.01
login	3	6.50	0.41	20.13	15.84	0.14	0.02
ps	8	6.33	0.76	10.50	8.37	0.09	0.07
3bc0	1	6.07	0.06	0.25	95.06	0.06	0.26
as2	8	6.05	0.51	4.58	11.80	0.06	0.11
list	2	5.91	0.54	4.02	11.02	0.27	0.13
front	1	5.73	0.08	0.22	73.11	0.08	0.36
ed	27	5.21	0.54	248.55	9.72	0.02	0.00
ntbl	1	4.51	0.08	5.13	53.56	0.08	0.02
back	1	3.96	0.13	0.47	30.09	0.13	0.28
3bas	1	3.89	0.05	0.38	73.38	0.05	0.14
sort	18	3.79	0.14	6.93	26.67	0.01	0.02
spell2	5	3.60	0.33	3.07	10.89	0.07	0.11
userdir	43	3.04	0.39	5.62	7.85	0.01	0.07
n3butil	2	2.81	0.04	69.45	73.33	0.02	0.00
col	5	2.47	0.23	24.60	10.54	0.05	0.01
pr	6	2.37	0.19	1.92	12.68	0.03	0.10
haspxmit	33	2.32	0.61	12.12	3.81	0.02	0.05
echo	106	2.32	0.39	14.67	6.02	0.00	0.03
tcpp	9	2.17	0.07	1.13	31.35	0.01	0.06
get	2	2.17	0.07	0.65	30.81	0.04	0.11
sed	9	2.02	0.10	2.63	20.89	0.01	0.04
delta	1	1.74	0.05	1.28	34.27	0.05	0.04
uuxqt	6	1.64	0.05	1.55	30.96	0.01	0.03
con	7	1.31	0.15	38.02	8.49	0.02	0.00
grep	6	1.26	0.20	1.48	6.21	0.03	0.14
admin	2	1.25	0.05	0.67	23.61	0.03	0.08
find	2	1.09	0.07	0.35	15.19	0.04	0.20
prt	6	1.06	0.07	1.70	14.35	0.01	0.04
rm	23	0.97	0.17	2.42	5.86	0.01	0.07
mail	5	0.97	0.08	5.72	12.78	0.02	0.01
103	5	0.87	0.11	0.82	7.62	0.02	0.14
npf	5	0.86	0.07	19.38	11.51	0.01	0.00
date	78	0.82	0.12	2.65	7.04	0.00	0.04
tcc	10	0.77	0.07	15.28	11.14	0.01	0.00

Appendix 2.1

expr	46	0.74	0.10	2.15	7.42	0.00	0.05
logdir	11	0.71	0.09	1.03	7.53	0.01	0.09
stty	36	0.56	0.07	1.83	8.06	0.00	0.04
a.out	8	0.54	0.06	1.22	8.99	0.01	0.05
spell0	5	0.52	0.07	1.13	7.09	0.01	0.06
pwd	9	0.46	0.14	2.15	3.38	0.02	0.06
who	23	0.45	0.08	1.60	5.46	0.00	0.05
unpack	1	0.44	0.03	0.72	14.58	0.03	0.04
gsi	2	0.42	0.07	10.47	5.75	0.04	0.01
gettriv	12	0.41	0.07	9.03	6.21	0.01	0.01
rbsh	1	0.40	0.03	28.93	12.02	0.03	0.00
3bm3	1	0.40	0.03	0.12	15.27	0.03	0.22
whoo	2	0.38	0.04	1.18	9.27	0.02	0.03
wc	21	0.34	0.06	1.55	5.79	0.00	0.04
getpi	12	0.31	0.04	4.63	7.97	0.00	0.01
cat	17	0.30	0.05	4.83	5.98	0.00	0.01
learn	2	0.30	0.02	57.42	14.37	0.01	0.00
tail	5	0.28	0.03	0.87	8.46	0.01	0.04
mv	6	0.26	0.02	0.27	12.21	0.00	0.08
lines	6	0.25	0.03	0.87	7.98	0.01	0.04
3bcc	2	0.25	0.02	2.50	13.94	0.01	0.01
bdiff	1	0.24	0.02	0.67	13.08	0.02	0.03
cc	4	0.22	0.02	9.87	8.49	0.01	0.00
cron	21	0.21	0.03	0.62	6.48	0.00	0.05
uulog	1	0.21	0.01	0.43	15.11	0.01	0.03
diff	1	0.21	0.02	0.30	12.54	0.02	0.05
mrrep	1	0.20	0.01	2.13	14.65	0.01	0.01
cp	5	0.19	0.06	0.38	3.37	0.01	0.15
dis	1	0.19	0.03	0.80	7.33	0.03	0.03
logname	10	0.18	0.02	0.40	7.36	0.00	0.06
deroff	5	0.17	0.02	0.40	10.03	0.00	0.04
spell3	5	0.16	0.02	3.25	8.89	0.00	0.01
lss	4	0.14	0.01	0.35	9.35	0.00	0.04
spell1	5	0.12	0.01	0.43	8.02	0.00	0.03
time	6	0.10	0.02	9.47	5.43	0.00	0.00
chmod	8	0.09	0.01	0.20	8.78	0.00	0.05
hasprecv	1	0.09	0.02	0.13	5.06	0.02	0.13
getty	7	0.07	0.02	1.37	3.13	0.00	0.02
lcount	2	0.05	0.01	4.33	8.08	0.00	0.00
opr	1	0.05	0.00	0.10	11.71	0.00	0.04
rmail	1	0.04	0.01	0.17	5.42	0.01	0.04
uuclean	1	0.04	0.00	0.10	8.44	0.00	0.04
cgetty	1	0.03	0.01	11.83	3.22	0.01	0.00
chown	1	0.02	0.00	0.07	8.94	0.00	0.04
uname	2	0.02	0.00	0.10	5.94	0.00	0.04
rmdir	1	0.02	0.01	0.07	3.20	0.01	0.10
sleep	3	0.02	0.00	45.10	7.37	0.00	0.00
basename	1	0.02	0.00	0.03	8.06	0.00	0.06
mkdir	1	0.01	0.00	0.02	6.10	0.00	0.13
logtty	1	0.01	0.00	0.02	6.83	0.00	0.10
acctswit	1	0.01	0.00	0.07	6.50	0.00	0.02
accton	1	0.01	0.00	0.05	5.67	0.00	0.03
mrstty	1	0.00	0.00	0.00	7.09	0.00	0.00
mrecho	1	0.00	0.00	0.00	3.45	0.00	0.00

Appendix 2.2

Sample Hourly Job Mix DATA

lhuxd — machine

jbmx0606.10am — time

date

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR
TOTALS	1411	1506.72	42.91	16105.34	35.12	0.03	0.00
nroff	13	696.96	15.39	136.05	45.30	1.18	0.11
c	1	167.11	0.88	40.73	188.83	0.88	0.02
3bld	10	103.14	1.95	14.00	52.88	0.20	0.14
bsh	155	49.96	1.71	12059.33	29.26	0.01	0.00
nspeak	4	37.07	0.46	12.13	79.82	0.12	0.04
ex	4	33.11	0.60	39.68	55.46	0.15	0.02
aspd	28	27.83	1.57	37.18	17.71	0.06	0.04
sgen	2	24.13	0.17	9.92	145.97	0.08	0.02
haspmain	7	20.95	0.29	417.30	71.36	0.04	0.00
haspqer	14	20.56	0.96	14.43	21.35	0.07	0.07
make	4	19.99	0.53	31.57	37.94	0.13	0.02
du	1	18.30	1.88	9.73	9.71	1.88	0.19
cpp	6	17.19	0.43	2.92	40.36	0.07	0.15
learn	5	16.49	0.19	691.52	85.90	0.04	0.00
pr_3bldp	2	14.72	0.26	16.92	57.40	0.13	0.02
ed	32	14.44	0.76	514.12	18.98	0.02	0.00
col	6	13.04	0.97	109.93	13.44	0.16	0.01
tcl	5	11.19	0.29	1.72	38.84	0.06	0.17
cl	3	11.17	0.36	2.33	31.35	0.12	0.15
sh	201	10.04	0.88	254.13	11.40	0.00	0.00
eat	36	9.91	1.46	4.23	6.81	0.04	0.34
as2	9	9.36	0.58	5.70	16.24	0.06	0.10
tc0	9	9.00	0.22	2.75	40.14	0.02	0.08
ld	6	8.71	0.35	4.32	24.94	0.06	0.08
tc2	1	8.66	0.26	1.23	33.71	0.26	0.21
c0	3	8.61	0.28	1.92	31.02	0.09	0.14
ftp	4	7.76	0.75	8.25	10.34	0.19	0.09
ls	50	7.22	0.50	5.92	14.58	0.01	0.08
e	36	6.80	0.69	192.58	9.81	0.02	0.00
uuclco	9	6.69	0.14	12.30	49.26	0.02	0.01
login	3	6.50	0.41	20.13	15.84	0.14	0.02
ps	8	6.33	0.76	10.50	8.37	0.09	0.07
3bc0	1	6.07	0.06	0.25	95.06	0.06	0.26
list	2	5.91	0.54	4.02	11.02	0.27	0.13
front	1	5.73	0.08	0.22	73.11	0.08	0.36
ntbl	1	4.51	0.08	5.13	53.56	0.08	0.02
back	1	3.96	0.01	0.47	285.28	0.01	0.03
3bas	1	3.89	0.05	0.38	73.38	0.05	0.14
sort	18	3.79	0.14	6.93	26.67	0.01	0.02
spell2	5	3.60	0.33	3.07	10.89	0.07	0.11
userdir	43	3.04	0.39	5.62	7.85	0.01	0.07
n3butil	2	2.81	0.02	69.45	131.42	0.01	0.00
pr	6	2.37	0.19	1.92	12.68	0.03	0.10
haspxmit	33	2.32	0.61	12.12	3.81	0.02	0.05
echo	106	2.32	0.39	14.67	6.02	0.00	0.03
tcpp	9	2.17	0.07	1.13	31.35	0.01	0.06
get	2	2.17	0.07	0.65	30.81	0.04	0.11
sed	9	2.02	0.10	2.63	20.89	0.01	0.04
delta	1	1.74	0.05	1.28	34.27	0.05	0.04
uuxqt	6	1.64	0.05	1.55	30.96	0.01	0.03
con	7	1.31	0.15	38.02	8.49	0.02	0.00
grep	7	1.28	0.20	7.15	6.29	0.03	0.03
admin	2	1.25	0.05	0.67	23.61	0.03	0.08
find	2	1.09	0.07	0.35	15.19	0.04	0.20
prt	6	1.06	0.07	1.70	14.35	0.01	0.04
rm	23	0.97	0.17	2.42	5.86	0.01	0.07
mail	5	0.97	0.08	5.72	12.78	0.02	0.01
103	5	0.87	0.11	0.82	7.62	0.02	0.14
npf	5	0.86	0.07	19.38	11.51	0.01	0.00

Appendix 2.3

date	78	0.82	0.12	2.65	7.04	0.00	0.04
tcc	10	0.77	0.07	15.28	11.14	0.01	0.00
expr	46	0.74	0.10	2.15	7.42	0.00	0.05
logdir	11	0.71	0.09	1.03	7.53	0.01	0.09
hqptape	1	0.59	0.03	5.43	18.03	0.03	0.01
stty	36	0.56	0.07	1.83	8.06	0.00	0.04
a.out	8	0.54	0.06	1.22	8.99	0.01	0.05
spell0	5	0.52	0.07	1.13	7.09	0.01	0.06
cat	19	0.48	0.07	89.73	6.84	0.00	0.00
pwd	9	0.46	0.14	2.15	3.38	0.02	0.06
who	23	0.45	0.08	1.60	5.46	0.00	0.05
unpack	1	0.44	0.03	0.72	14.58	0.03	0.04
gsi	2	0.42	0.07	10.47	5.75	0.04	0.01
gettriv	12	0.41	0.07	9.03	6.21	0.01	0.01
rbsh	1	0.40	0.03	28.93	12.32	0.03	0.00
3bm3	1	0.40	0.03	0.12	15.27	0.03	0.22
whoo	2	0.38	0.04	1.18	9.27	0.02	0.03
wc	21	0.34	0.06	1.55	5.79	0.00	0.04
getpl	12	0.31	0.04	4.63	7.97	0.00	0.01
cc	5	0.30	0.02	13.27	12.41	0.00	0.00
tail	5	0.28	0.03	0.87	8.46	0.01	0.04
mv	6	0.26	0.02	0.27	12.21	0.00	0.08
lines	6	0.25	0.03	0.87	7.98	0.01	0.04
3bcc	2	0.25	0.01	2.50	28.79	0.00	0.00
bdiff	1	0.24	0.02	0.67	13.08	0.02	0.03
cron	21	0.21	0.03	0.62	6.48	0.00	0.05
uulog	1	0.21	0.01	0.43	15.11	0.01	0.03
diff	1	0.21	0.02	0.30	12.54	0.02	0.05
mrrep	1	0.20	0.01	2.13	19.00	0.01	0.00
cp	5	0.19	0.06	0.38	3.37	0.01	0.15
dis	1	0.19	0.03	0.80	7.33	0.03	0.03
logname	10	0.18	0.02	0.40	7.36	0.00	0.06
deroff	5	0.17	0.02	0.40	10.03	0.00	0.04
spell3	5	0.16	0.02	3.25	8.89	0.00	0.01
lss	4	0.14	0.01	0.35	9.35	0.00	0.04
spell1	5	0.12	0.01	0.43	8.02	0.00	0.03
time	6	0.10	0.02	9.47	5.43	0.00	0.00
cgetty	4	0.09	0.01	926.95	10.46	0.00	0.00
chmod	5	0.09	0.01	0.20	8.78	0.00	0.05
hasprecv	1	0.09	0.02	0.13	5.06	0.02	0.13
getty	7	0.07	0.02	1.37	3.13	0.00	0.02
lcount	2	0.05	0.01	4.33	8.08	0.00	0.00
opr	1	0.05	0.00	0.10	11.71	0.00	0.04
rmall	1	0.04	0.01	0.17	5.42	0.01	0.04
uuclean	1	0.04	0.00	0.10	8.44	0.00	0.04
sleep	4	0.03	0.00	60.13	12.85	0.00	0.00
chown	1	0.02	0.00	0.07	8.94	0.00	0.04
uname	2	0.02	0.00	0.10	5.94	0.00	0.04
rmdir	1	0.02	0.01	0.07	3.20	0.01	0.10
basename	1	0.02	0.00	0.03	8.06	0.00	0.06
mkdir	1	0.01	0.00	0.02	6.10	0.00	0.13
logtty	1	0.01	0.00	0.02	6.83	0.00	0.10
acctswit	1	0.01	0.00	0.07	6.50	0.00	0.02
accton	1	0.01	0.00	0.05	5.67	0.00	0.03
mrstty	1	0.00	0.00	0.00	7.08	0.00	0.00
mrecho	1	0.00	0.00	0.00	3.45	0.00	0.00

Appendix 2.4

Jun 7 10:06 1979 templ Page 1

Processes initiated for this hour

DATA FOR llpsl
dated : jbmx0606(1) 10am

Total Cpu For this Hour = 34.810001
Total Number of Commands = 2439.000000

Sample Categorized Job Mix Report

	TCPU	TSIZE	NUMCMD	CPU(%)
db :				
diff	0.640	19.340	7.000	1.839
grep	2.820	6.520	277.000	8.101
sort	0.230	31.610	4.000	0.661
bdiff	0.090	13.850	3.000	0.259
unpack	0.020	14.700	1.000	0.057
Total Sum	3.800	86.020	292.000	10.916
doc :				
nroff	2.800	44.150	9.000	8.044
ed	0.930	11.460	21.000	2.672
sed	0.110	15.690	21.000	0.316
spell2	0.130	10.590	1.000	0.373
e	0.350	11.560	13.000	1.005
col	0.020	7.260	1.000	0.057
spell0	0.030	6.940	1.000	0.086
tbl	0.250	51.580	4.000	0.718
Total Sum	4.620	159.230	71.000	13.272
io :				
pr	0.030	13.850	3.000	0.086
mv	0.010	7.670	3.000	0.029
cat	0.200	4.910	36.000	0.575
cp	0.110	4.300	30.000	0.316
cpx	0.340	8.510	3.000	0.977
line	0.740	6.860	260.000	2.126
tr	0.010	7.160	6.000	0.029
hqptape	0.080	15.990	1.000	0.230
cpio	0.040	17.290	1.000	0.115
df	0.440	6.310	5.000	1.264
du	0.410	8.150	2.000	1.178
Total Sum	2.410	101.000	350.000	6.923
nw :				
ftp	0.350	10.210	10.000	1.005
haspmain	1.250	6.290	6.000	3.591
haspqer	0.790	20.940	17.000	2.269
haspxmit	0.320	3.840	25.000	0.919
rjestat	0.010	12.330	2.000	0.029
tssjob	0.560	22.160	15.000	1.609
con	0.720	9.770	9.000	2.068

Appendix 3.1

bscsend	0.230	6.240	2.000	0.661
dmcsend	0.080	5.650	2.000	0.230
bscread	0.080	6.170	2.000	0.230
bin	C.020	17.560	1.000	0.057
dmcread	0.010	7.920	2.000	0.029
uucico	0.110	49.800	9.000	0.316
uuxqt	0.040	29.030	6.000	0.115
hasprecv	0.020	4.990	1.000	0.057
binq	0.010	9.930	2.000	0.029
bin.psl.	2.030	118.140	2.000	5.832
binr	0.110	17.440	6.000	0.316
sendlog	0.100	9.430	1.000	0.287
lines	0.030	7.730	6.000	0.086
aspswapj	0.180	17.280	4.000	0.517
asploadj	0.190	28.710	12.000	0.546
Total Sum	7.240	421.560	142.000	20.799
sgs :				
c1	0.340	31.410	3.000	0.977
as2	0.290	11.610	3.000	0.833
cpp	0.300	36.070	3.000	0.862
c0	0.290	31.180	3.000	0.833
cc	0.020	8.010	3.000	0.057
swap3.4	0.090	7.860	27.000	0.259
Total Sum	1.330	126.140	42.000	3.821
usr :				
bsh	4.480	9.730	142.000	12.870
sh	1.340	10.380	299.000	3.849
echo	0.820	6.600	240.000	2.356
ls	0.810	14.890	54.000	2.327
rm	0.290	4.830	43.000	0.833
date	0.230	7.560	118.000	0.661
chmod	0.060	8.690	28.000	0.172
expr	0.140	8.030	57.000	0.402
chown	0.010	7.510	3.000	0.029
pwd	0.080	4.160	12.000	0.230
wc	0.040	5.710	14.000	0.115
ps	0.770	8.410	11.000	2.212
userdir	0.450	7.720	78.000	1.293
mail	0.740	12.710	51.000	2.126
env	0.120	9.140	13.000	0.345
login	0.010	11.090	1.000	0.029
logdir	0.040	5.870	21.000	0.115
time	0.020	5.940	6.000	0.057
who	0.020	5.660	6.000	0.057
ln	0.010	5.250	2.000	0.029
Total Sum	10.480	160.780	1199.000	30.106

sys :

Appendix 3.2

getty	0.070	3.330	24.000	0.201
gettriv	0.070	6.140	12.000	0.201
getpl	0.040	7.820	12.000	0.115
cron	0.050	6.950	34.000	0.144
cgetty	0.010	3.340	2.000	0.029
uulog	0.010	14.450	1.000	0.029
stty	0.090	8.770	44.000	0.259
logname	0.060	8.070	37.000	0.172
uname	0.010	9.010	5.000	0.029
pdtsh	0.090	22.490	1.000	0.259
joblg	0.090	10.980	21.000	0.259
Total Sum	0.590	101.350	193.000	1.695
misc :				
eat	1.390	6.810	36.000	3.993
Total Sum	1.390	6.810	36.000	3.993
uni :				
sp	0.040	5.650	5.000	0.115
Total Sum	0.040	5.650	5.000	0.115
cms :				
delta	0.160	36.090	2.000	0.460
m2find	0.530	53.900	9.000	1.523
edput	0.060	26.540	2.000	0.172
get	0.580	31.060	14.000	1.666
sget	0.060	26.470	2.000	0.172
cont	0.450	5.520	3.000	1.293
edget	0.160	27.410	4.000	0.460
evalparm	0.210	15.360	14.000	0.603
ped	0.490	18.790	10.000	1.408
hash	0.020	8.980	10.000	0.057
row	0.010	12.660	1.000	0.029
submit	0.050	29.930	3.000	0.144
mrmap	0.020	52.350	2.000	0.057
usrbin	0.010	8.360	1.000	0.029
usrname	0.010	8.680	1.000	0.029
usracct	0.010	9.090	1.000	0.029
swapfix	0.020	13.750	1.000	0.057
chip3	0.010	6.340	1.000	0.029
evalfix	0.010	14.610	1.000	0.029
notify	0.010	62.130	8.000	0.029
Total Sum	2.880	468.020	90.000	8.273

Appendix 3.3

Jun 7 09:58 1979 temp1 Page 1

DATA FOR Ihps1
dated : Ibmx0606.10am — Processes existing for this hour

Total Cpu For this Hour = 34.880001
Total Number of Commands = 2491.000000

Sample Categorized Job Mix Report

	TCPU	TSIZE	NUMCMD	CPU(%)
db :				
diff	0.640	19.340	7.000	1.835
grep	2.820	6.540	278.000	8.085
sort	0.230	31.610	4.000	0.659
bdiff	0.090	13.850	3.000	0.258
unpack	0.020	14.700	1.000	0.057
Total Sum	3.800	86.040	293.000	10.894
doc :				
nroff	2.910	51.260	10.000	8.343
ed	0.680	19.880	22.000	1.950
sed	0.110	15.690	21.000	0.315
spell2	0.130	10.590	1.000	0.373
e	0.290	13.770	13.000	0.831
col	0.020	7.260	1.000	0.057
spell0	0.030	6.940	1.000	0.086
tbl	0.250	51.580	4.000	0.717
Total Sum	4.420	176.970	73.000	12.672
io :				
pr	0.020	19.290	3.000	0.057
mv	0.010	7.670	3.000	0.029
cat	0.200	4.910	36.000	0.573
cp	0.110	4.300	30.000	0.315
cpx	0.340	8.510	3.000	0.975
line	0.740	6.860	260.000	2.122
tr	0.010	7.160	6.000	0.029
hoptape	0.080	15.990	1.000	0.229
cpio	0.040	17.290	1.000	0.115
df	0.440	6.310	5.000	1.261
du	0.410	8.150	2.000	1.175
Total Sum	2.400	106.440	350.000	6.881
nw :				
ftp	0.350	10.210	10.000	1.003
haspmain	0.230	34.080	7.000	0.659
haspqer	0.790	20.940	17.000	2.265
haspxmit	0.320	3.840	25.000	0.917
rjestat	0.010	12.330	2.000	0.029
tssjob	0.560	22.160	15.000	1.606
con	0.670	10.410	9.000	1.921

Appendix 3.4

bscsend	0.490	24.770	3.000	1.405
dmcsend	0.100	15.450	3.000	0.287
bscread	0.130	21.300	3.000	0.373
bin	0.020	17.560	1.000	0.057
dmcread	0.020	21.380	3.000	0.057
uucico	0.110	49.800	9.000	0.315
uuxqt	0.040	29.030	6.000	0.115
hasprecv	0.020	4.990	1.000	0.057
binq	0.010	9.930	2.000	0.029
bin.psl.	3.170	412.030	3.000	9.088
binr	0.110	17.440	6.000	0.315
sendlog	0.100	9.430	1.000	0.287
lines	0.030	7.730	6.000	0.086
aspswapj	0.180	17.480	5.000	0.516
aspI oadj	0.190	28.710	12.000	0.545
Total Sum	7.650	801.000	149.000	21.932

sgs :

c1	0.340	31.410	3.000	0.975
as2	0.290	11.610	3.000	0.831
cpp	0.300	36.070	3.000	0.860
c0	0.290	31.180	3.000	0.831
cc	0.020	8.010	3.000	0.057
swap3.4	0.090	7.860	27.000	0.258
Total Sum	1.330	126.140	42.000	3.813

usr :

bsh	4.310	15.020	165.000	12.357
sh	1.360	11.280	303.000	3.899
echo	0.820	6.620	241.000	2.351
ls	0.810	14.890	54.000	2.322
rm	0.290	4.830	43.000	0.831
date	0.230	7.560	118.000	0.659
chmod	0.060	8.690	28.000	0.172
expr	0.140	8.030	57.000	0.401
chown	0.010	7.510	3.000	0.029
pwd	0.080	4.160	12.000	0.229
wc	0.040	5.710	14.000	0.115
ps	0.770	8.410	11.000	2.208
userdir	0.450	7.720	78.000	1.290
mail	0.740	12.710	51.000	2.122
env	0.120	9.140	13.000	0.344
login	0.030	70.710	3.000	0.086
logdir	0.040	5.870	21.000	0.115
time	0.020	5.940	6.000	0.057
who	0.020	5.660	6.000	0.057
ln	0.010	5.250	2.000	0.029
Total Sum	10.350	225.710	1229.000	29.673

sys :

Appendix 3.5

getty	0.070	3.410	26.000	0.201
gettriv	0.070	6.140	12.000	0.201
getpi	0.040	7.820	12.000	0.115
cron	0.050	6.950	34.000	0.143
cgetty	0.010	3.650	3.000	0.029
uulog	0.010	14.450	1.000	0.029
stty	0.090	8.770	44.000	0.258
logname	0.060	8.070	37.000	0.172
uname	0.010	9.010	5.000	0.029
pdtsh	0.010	432.320	2.000	0.029
joblg	0.090	10.980	21.000	0.258
Total Sum	0.510	511.570	197.000	1.462
misc :				
eat	1.390	6.810	36.000	3.985
Total Sum	1.390	6.810	36.000	3.985
uni :				
sp	0.040	5.650	5.000	0.115
Total Sum	0.040	5.650	5.000	0.115
cms :				
delta	0.270	36.480	3.000	0.774
m2find	0.530	53.900	9.000	1.519
edput	0.080	35.570	3.000	0.229
get	0.550	34.990	15.000	1.577
sget	0.090	30.170	3.000	0.258
cont	0.550	6.010	4.000	1.577
edget	0.100	43.270	4.000	0.287
evalparm	0.210	15.360	14.000	0.602
ped	0.390	24.260	11.000	1.118
hash	0.020	8.980	10.000	0.057
row	0.010	12.660	1.000	0.029
submit	0.050	29.930	3.000	0.143
mrmap	0.020	52.350	2.000	0.057
usrbin	0.010	8.360	1.000	0.029
usrname	0.010	8.680	1.000	0.029
usracct	0.010	9.090	1.000	0.029
swapfix	0.020	13.750	1.000	0.057
chip3	0.010	6.340	1.000	0.029
evalfix	0.010	14.610	1.000	0.029
notify	0.010	62.130	8.000	0.029
Total Sum	2.950	506.890	96.000	8.458

Appendix 3.6

Appendix 4.1

<u>Job Mix Categorization</u>

<u>DATA BASE</u>

DATA BASE (db)

```
diff
grep
sort
bdiff
cut
unpack
tail
egrep
split
file
join
cmp
dd
touch
size
sum
awk
bsplit
uniq
sdiff
cpall
read
readtm
nlsort
null
```

Documentation (doc)

```
nroff
npf
ex
ed
sed
spell2
e
col
spell0
deroff
spell1
spell3
tbl
typo
troff
roff
xed
bfs
ex1
ex2
neqn
reform
eqn
ptx
graph
null
```

Input/Output (io)

```
opr
pr
mv
cat
cp
```

Appendix 4.2

nm
spool
cpx
line
gath
tr
hqptape
banner
nspool
cpio
df
cu
du
tee
spool2
crypt
diskusag
Print
tp
450
gsi
oprl
abtape
dtee
fd2
spooless
tcpio

Input/Output (io)

---

null
haspdisp
ftp
aspd
uuclean
send
haspmain
haspqer
haspxmit
haspinit
rjestat
tssjob
con
bin.psi
bscsend
dmcsend
bscread
bin
dmcread
uucico
uuxqt
bscabort
dmcabort
hasprecv
binq
bin.psi.
binr
sendlog
usend

Network (nw)

Appendix 4.3

Network (nw)

hasphalt
fsend
conclr
cptss
lines
tss
aspswapj
haspcopy
asploadj
asj
xmittss

null

Software Generation Tools (sgs)

c1
3bdis
dmpp
tcpp
ncpp
swapjob
as2
cpp
c0
ld
m8ld
tcc
m8list
m8asl
m8asmb
cc
tc0
m8cpp
m8hx
tccp
m8as
tc1
adb
ncc
nc0
nc1
ar
swap3.4
spitbol
strip
swap3.1c
swap.scr
swap.1c
lint
cb
tc2
3bld
3bc0
3bas
3bm3
pr_3bldp
c2
yacc
3bcc

Appendix 4.4

Software Generation Tools (sgs)

list
dis
lister
tlint1
tlint2
lex
br_3bldp
bas
pr_3bsge
m68ccc
h2pc
end
phend
as
list.pro
m8reloc
m68cpp
m68as.cc
fixldp
cechart
cxref
m4
micat
nc2
pargen
tyacc
ldp
b16cc
b16dis
b16list
db
m8split
m8fod
nm8ol
m8patch
m8image
nm8ld
m8comp
m8parity
p8cc
m8crc
m8block
m8bugsta
m8sag
m8nm
m8prom
m8cc
cdb
fcl
cclo.cc.
3bldp
newfixld
onlint
cref
null
bsh

User (usr)

User (usr)

```
nspeak
tty
sh
echo
ls
rm
date
chmod
expr
chown
chgrp
pwd
wc
rmdir
cd
ps
userdir
mail
env
login
logdir
time
who
kill
whoo
passwd
find
nfind
mkdir
tabs
basename
write
quiz
rmall
learn
dc
bc
mesg
esh
od
craps
ttt
startrek
reversi
bj
tsh
-bsh
rbsh
mount
umount
rmail
nohup
thredit
hascii
mcexec
ectdump
```

Appendix 4.5

Appendix 4.6

User (usr)

ccwrite
ccread
obsh
othello
rmdel
blackjac
gcat
vowels
cal
chguser
dsw
hamurabi
help
hexdump
ln
masrdms
masrsmc
mass
maswrap
mvall
nls
test
whatsnew
tfind
monitor
moo
chess3
nice
maswrdp
masoflwr
masoflrd
checksum
rmtree
prof

System (sys)

null
acctmerg
init
logtty
getty
gettriv
getpi
cron
cgetty
uulog
stty
setty
logname
acctwtmp
accton
acctswit
uname
pdtsh
joblg
makekey
su
phinit

Appendix 4.7

```
acctprc1
acctcon1
acctcom
sync
jobsync
acctcms
acctcom2
nt
clock
testcom2
dump
datar
acctprc1
```

System (sys)

```
null
eat
sleep
tuop
3butil
n3butil
```

Miscellaneous (misc)

```
null
userline
cpchip.N
bwm.cs
compact
atd
shqer
a.out
lscols
mon
nmon
sp
swcont
fmtuser
spit
work
xcrypt
actmr
blocker
input
sconvert
setup
fixshipn
vc10
labelit
p
lpass1
lpass2
toscans3
utoo
otou
lcount
lock
s
start.csgs
ditD
sum.cs
```

Unidentified (uni)

start.c
atf
atr
att
mhandler
lsort
tapein
cnvert2
redo.cs
chknum
index
crt
a
zap
103
ndoc
sgen
prnsyms
sgenmp
comp
xpass
dotss
tujs
format
bls
dirs
junk
st
xpcex
explist
sp2
ti
ce
105
binpack
phdr
U
wrtails
L
readofl
mswrsar
bgnblk
extract
wide
endblk
[
begin
txcmd
mrets
am0
chg
mlst
dtjump
ramit
symbol
Lx

Unidentified (uni)

Appendix 4.8

Appendix 4.9

```
aststst
clst
nameparm
adm
dtdest
temp.p5.
temp.p4.
temp.p8.
temp.p10
genrad
temp.p1.
temp.p7.
mswradr
romck
pcload
pre.mpre
yarp
miro
mdef
wcspost
dec
cdf
moduser
excofl
setarg
scrsetup
erlytrm
setmode
rtnstat
wrtmijs
quit
number.o
ntbl
newer
addept
emul1
emul3
label
daystats
weekstat
binhex
front
back
cube
colform
dmpr
toscans2
paste
c
bob
joe
lem
newids
ctoh
opt
sects
```

Unidentified (uni)

Appendix 4.10

Unidentified (uni)

trop
twc
?
mrc345:f
splinter
lastpgm
dcm
msects
rf2b
pikp
snap
1
pcdgnid
op
hemi
h2i
smal2
sc
soxpnd
ectgen
slave
onlrst
endonl
bgnonl
expand
mchpost
onmspost
temp.pi2
tuops
clrpass
mc
hello
crpost
squash
.glprt

Change Management Tools (cms)

null
delta
finddir
m2find
m2edit
assign
edput
get
sget
cont
edget
evalparm
mrdriver
create
ped
hash
loadjob
row
submit
pjlog
mrnumgen

Appendix 4.11

Change Management tools (cms)

schednxt
approve
cow
pget
sched
ptchsp
mrmap
usrbin
usrname
usracct
swapfix
mrchange
admin
pdelta
asusr
nameparms
pprt
gparm
gs
ditD
m2editD
2e
cnvert2b
killm2
mrsort
dropcnt
make
tmake
edit
mrstat
3bcreate
lss
mrinstat
nmake
mrecho
mrstty
mrrep
build
mredit
oldmr
merge
addcr
chghist
mkbootf
prt
chip3
3e
evalfix
3bget
3bupdate
3bdelta
imomtr
imcnttr
notify
patchsp
reject

```
what
hashsum
3bfrs
3bprt
edpag
null
```

Change Management Tools (cms)

Appendix 4.12

UNI Category Percentages - UNIX C

time	0604 (% CPU)	0605 (% CPU)	0606 (% CPU)
5am	18.13[1]	.517	15.58[8]
6am	1.584	.112	19.653[9]
7am	2.756	0	11.111[10]
8am	21.141[2]	10.675[5]	10.444[11]
9am	14.091[3]	3.783	4.165
10am	7.175	7.555	1.646
11am	5.466	1.203	6.844
12pm	8.274	1.001	1.068
1pm	4.528	1.422	5.937
2pm	5.078	3.742	1.087
3pm	2.667	5.383	1.184
4pm	11.503[4]	10.125[6]	.86
5pm	7.923	10.650[7]	.6
6pm	0	.626	0

1 daystats, CPU = 1.85 min
2 xcrypt, CPU = 5.51 min
3 wrtmiis, CPU = 1.04 min
4 number.o; CPU = 3.46 min
5 xcrypt, CPU = 1.1 min
6 pre.mpre + wrtmiis, CPU = 2.84 min
7 mlst, CPU = 3.23 min
8 ectgen, CPU = 1.8 min
9 ectgen, CPU = 1.79 min
10 all commands < 1 min
11 symbol, CPU = 1.8 min

Appendix 5.1

UNI Category Percentages - UNIX D

time	0604 (%CPU)	0605 (%CPU)	0606 (%CPU)
5am	12.43[1]	.553	1.044
6am	4.866	4.937	3.018
7am	0	0	1.628
8am	.355	.265	5.334
9am	.969	2.018	13.096[3]
10am	4.068	28.525[2]	3.263
11am	.124	6.815	4.304
12pm	3.836	.573	3.157
1pm	.868	2.625	4.185
2pm	.130	4.93	3.479
3pm	1.129	2.757	1.1
4pm	6.612	.255	4.353
5pm	1.587	.667	8.011
6pm	.0	1.624	1.984

[1] daystats, CPU = 1.91 min
e, CPU = 12.18 min
c, CPU = 5.57 min

UNI Category Percentages - IHPS1

time	0604 (% CPU)	0605 (% CPU)	0606 (% CPU)
5am	0	0	0
6am	0	0	0
7am	0	0	0
8am	.08	.124	.509
9am	17.45[1]	.027	.102
10am	26.945[2]	0	.115
11am	13.005[3]	.032	.033
12pm	19.563[4]	.718	.078
1pm	5.471	1.726	.028
2pm	1.328	.901	.164
3pm	1.24	.148	1.195
4pm	.112	0	2.216
5pm	.541	0	6.456
6pm	0	0	7.396

[1] a.out, CPU = 5.82 min
[2] a.out, CPU = 8.55 min
[3] vc10, CPU = 2.37 min
[4] vc10, CPU = 3.65 min

UNI Category Percentages - IHPS2

time	0604 (% CPU)	0605 (% CPU)	0606 (% CPU)
5am	0	0	0
6am	0	0	0
7am	0	.49	9.152
8am	2.072	.713	8.614
9am	5.461	.512	4.696
10am	1.906	11.235[2]	11.045[6]
11am	4.565	76.071[3]	4.622
12pm	17.022[1]	23.297[4]	.239
1pm	7.91	5.176	7.867
2pm	2.141	3.16	5.277
3pm	8.428	9.535	9.719
4pm	7.515	27.259[5]	2.664
5pm	0	0	6.486
6pm	2.845	0	13.357[7]

[1] vel0, CPU = 4.7 min
[2] a.out, CPU = 1.91 min
[3] a.out, CPU = 42.03 min
[4] a.out, CPU = 3.09 min
[5] start.c, CPU = 8.15 min
[6] all commands < 1 min
[7] all commands < 1 min

Appendix 5.4

Commands Used By PDAEMON

command	freq/pdaemon	CPU sec/daemon	CPU sec/hr
date	1	.1	.4
echo	5	.5	2.0
who	1	.1	.4
wc	1	.1	.4
ps	1	.8	32.0
e	1	.2	.8
time	2	.2	.8
cp	1	.9	3.6
cc	1	24	96.0
sleep	1	.1	.4
			2.28 min

→ 2.28 CPU minutes/hr used by pdaemon collection

Appendix 6

Effect of Job Mix Tools

0604 - UNIX D

hr	total CPU (min)	% CPU
4am	1.63	7.9
10am	3.52	10.1
12pm	3.04	11.5
1pm	1.51	5.5
2pm	1.64	7.1
5pm	3.94	7.9

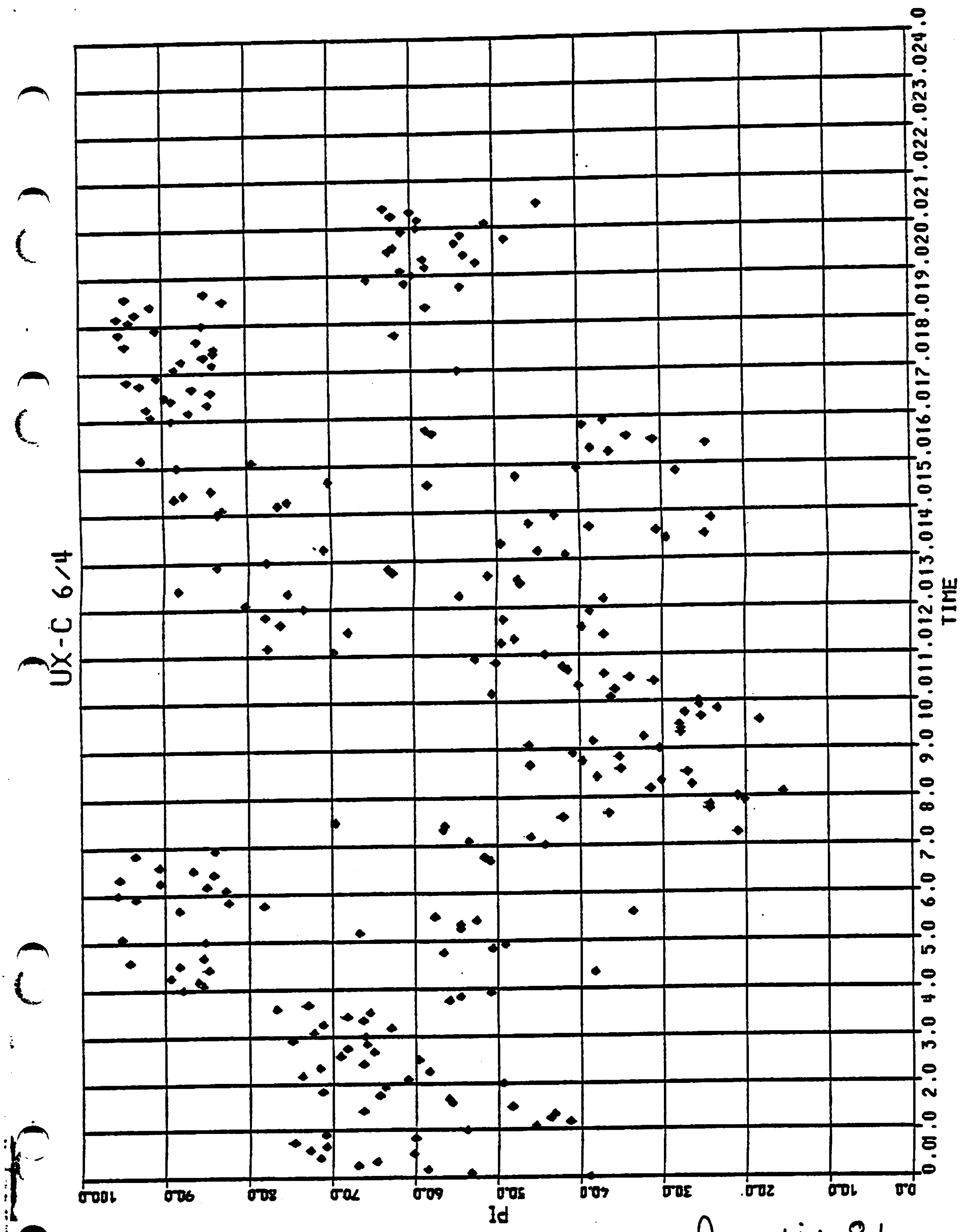
0605 - UNIX D

hr	total CPU (min)	% CPU
10am	.92	2
11am	1.17	2.3
1pm	3.72	12.1
2pm	1.68	5.6
3pm	2.61	6.4
4pm	.43	1.2
5pm	1.7	3.8

0606 - UNIX D

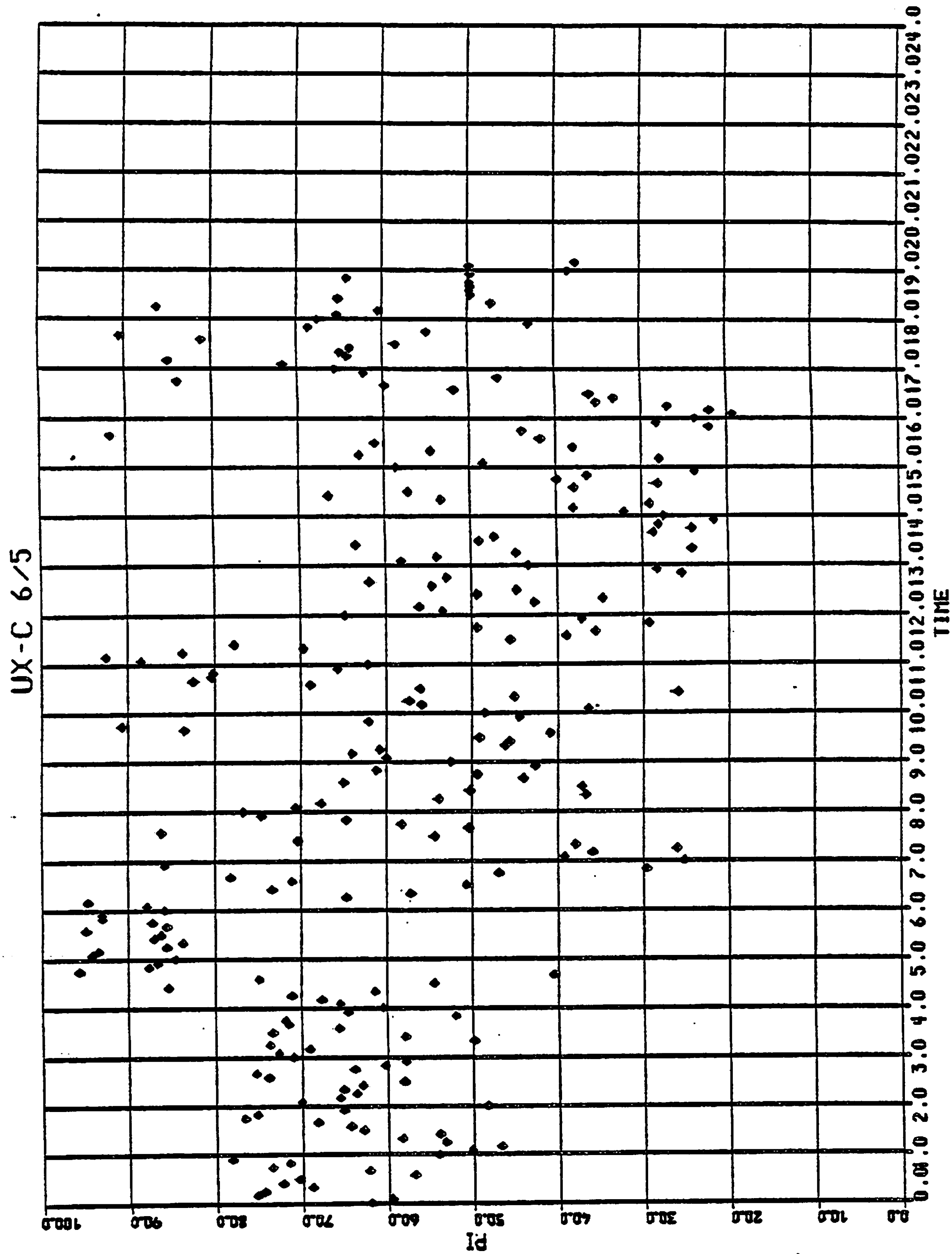
hr	total CPU (min)	% CPU
9am	1.14	2.5
1pm	3.91	10.1

9

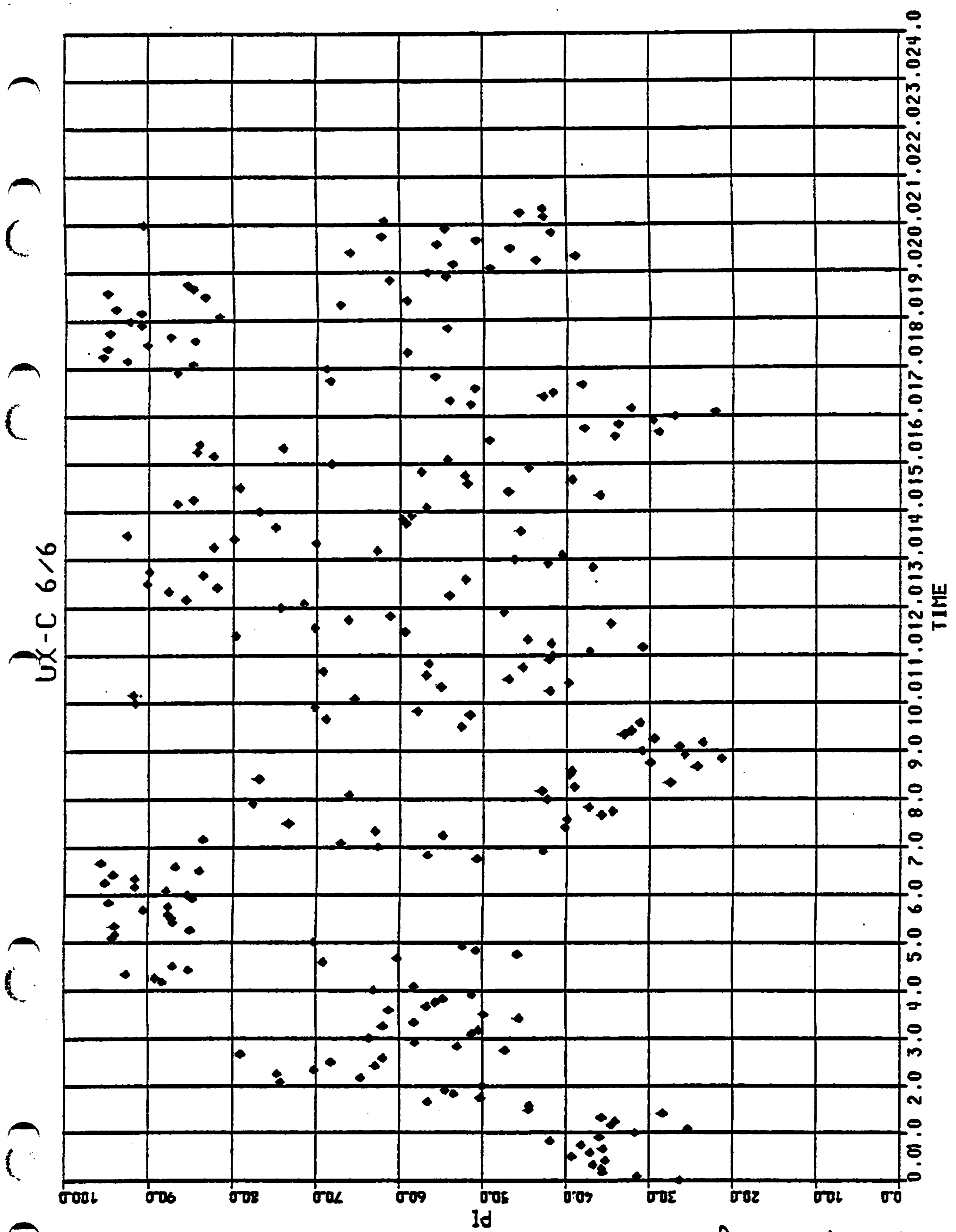


Appendix 8.1

JOB NAME=JLP***** TSS NO=0138 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=10:[illegible] PAGE [illegible]

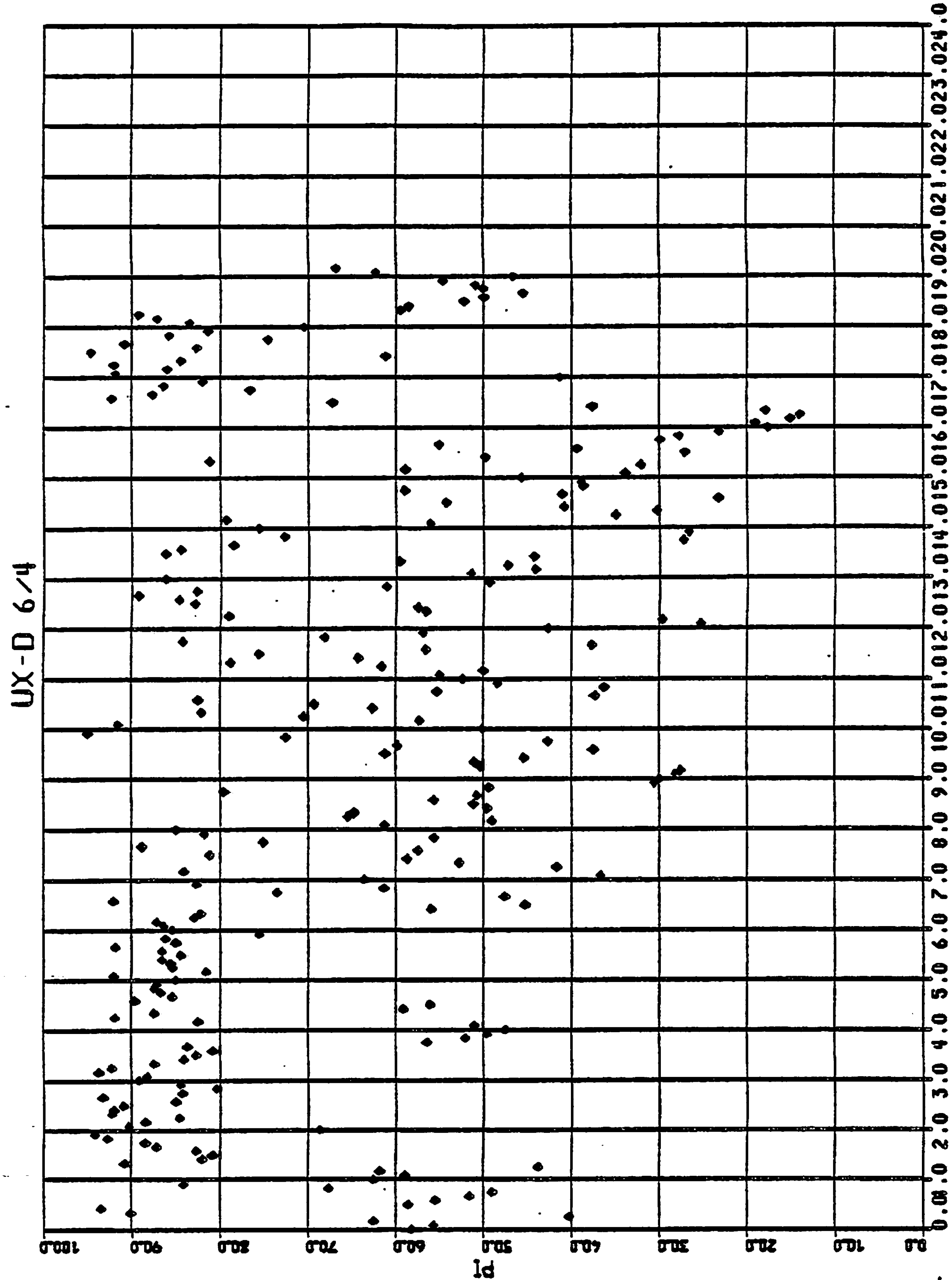


Appendix 8.2



Appendix 8.3

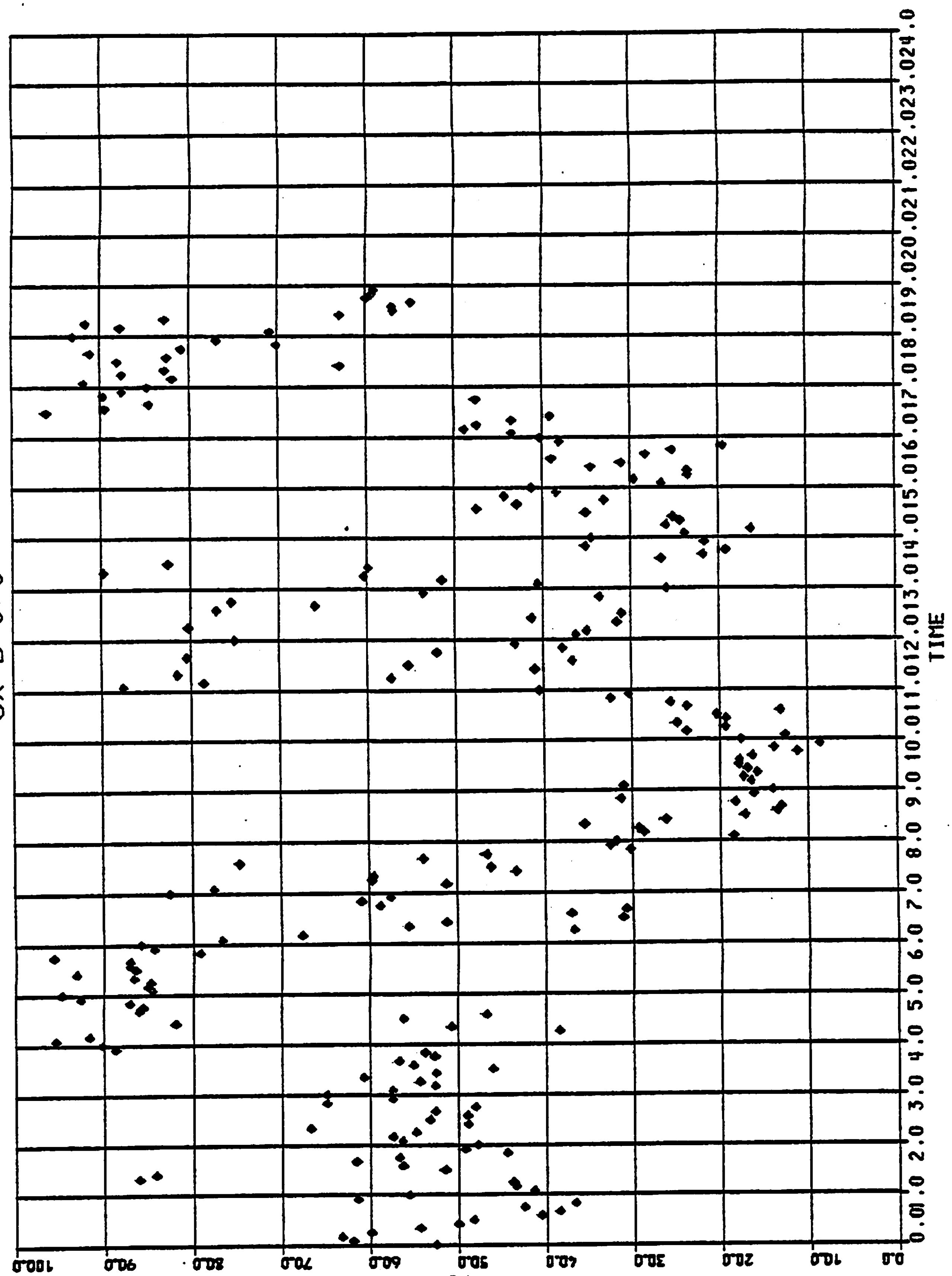
JOB NAME=JLP***** TSS NO=01C2 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=10:42:17



Appendix 8.4

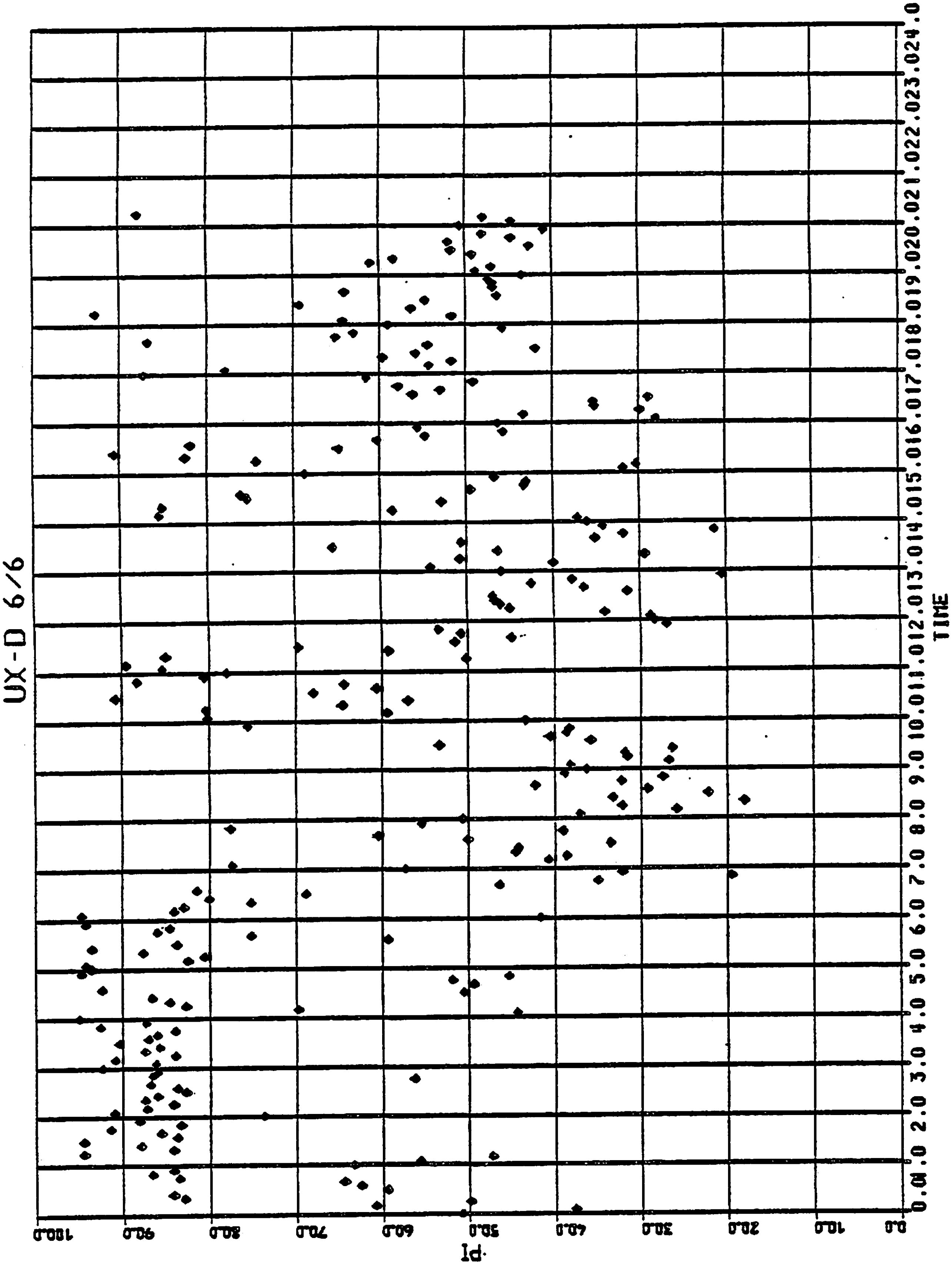
JOB NAME=JLF***** TSS NO=01AB BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=10:33:23 PAGE NO 001

JX-D 6/5



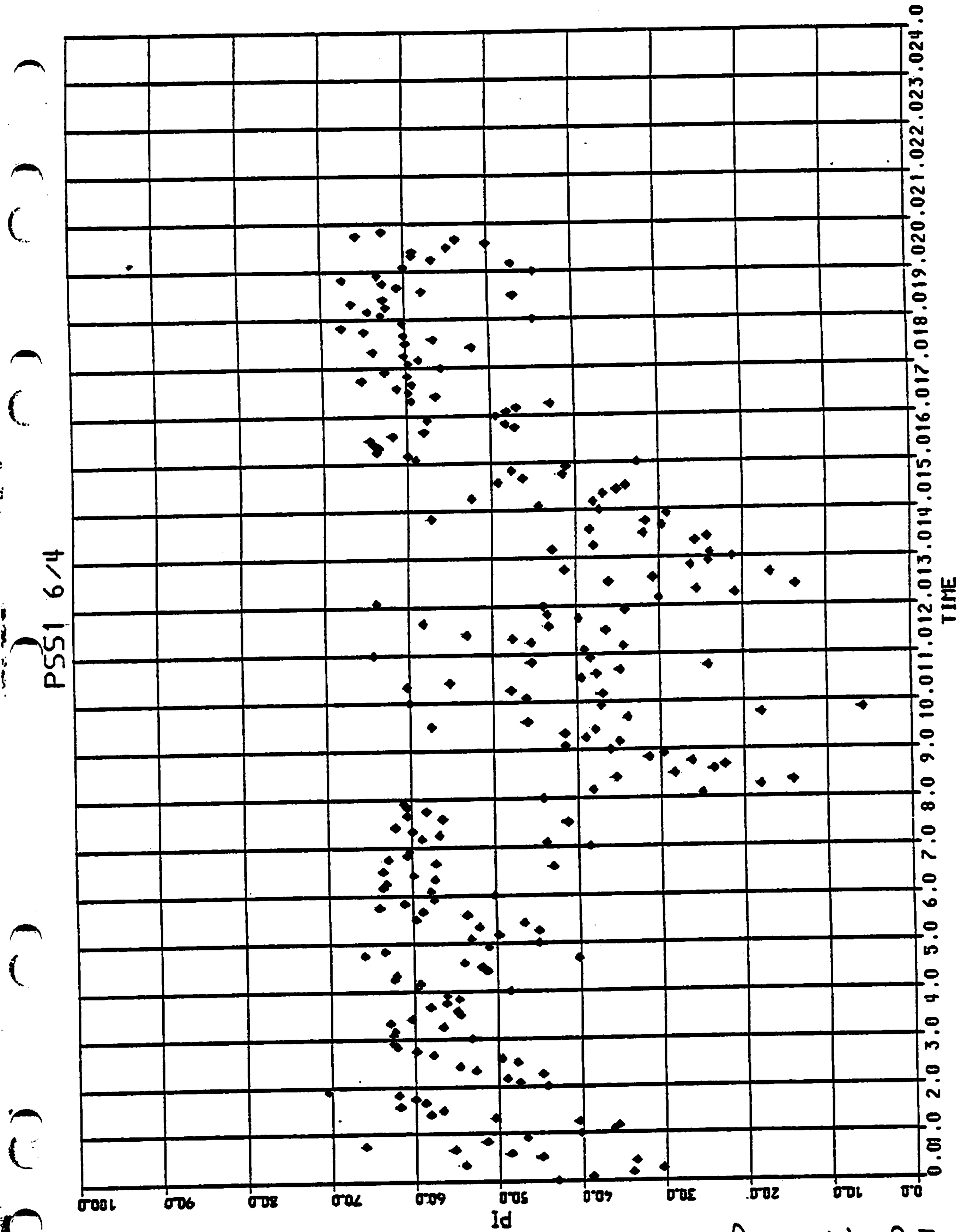
Appendix 8.5

JOB NAME=JLP***** TSS NO=01B0 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=10:35:14 PAGE NO 001



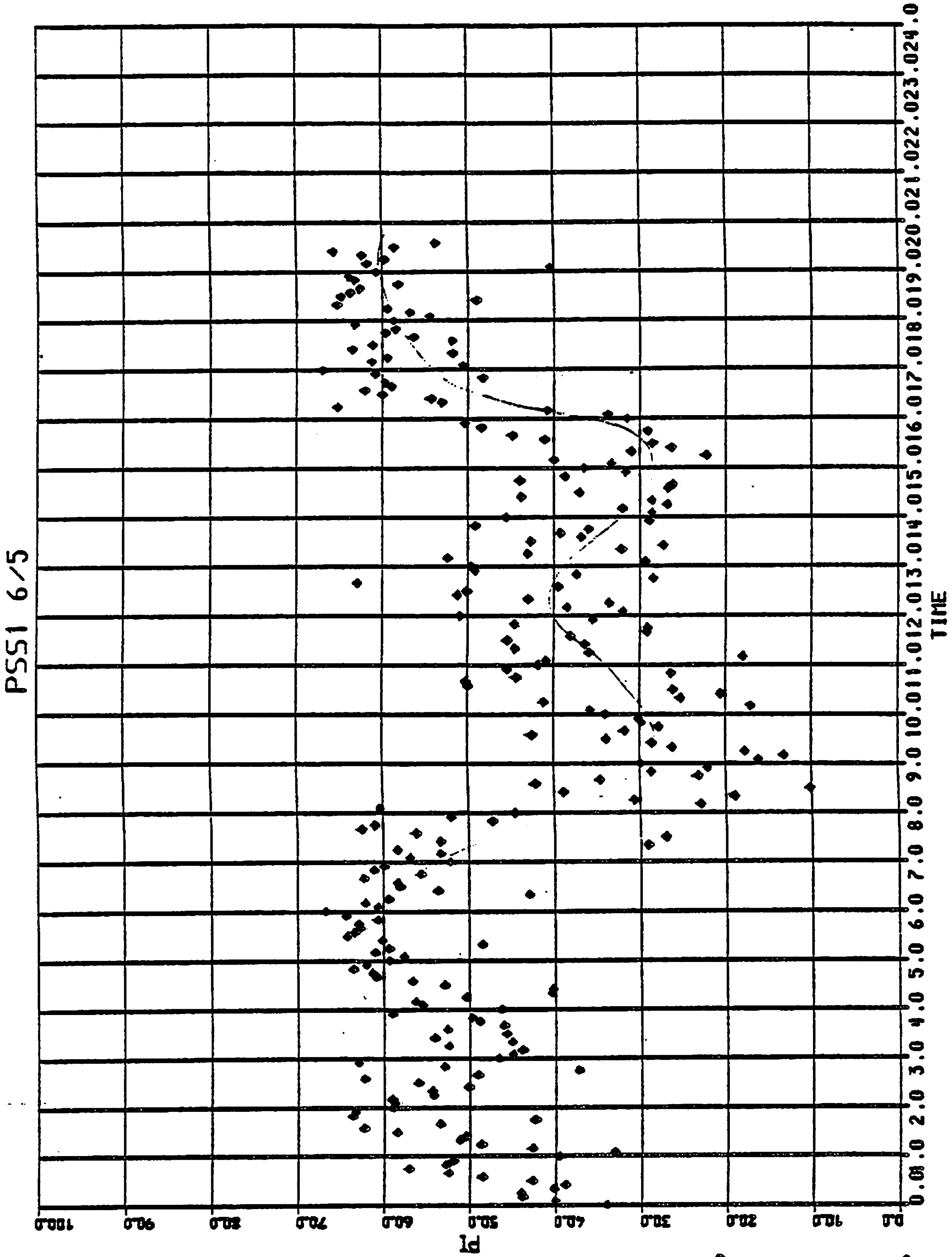
Appendix 8.6

JOB NAME=JLP***** TSS NO=0183 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/03/79 TIME=10:36:45 PAGE NO 001



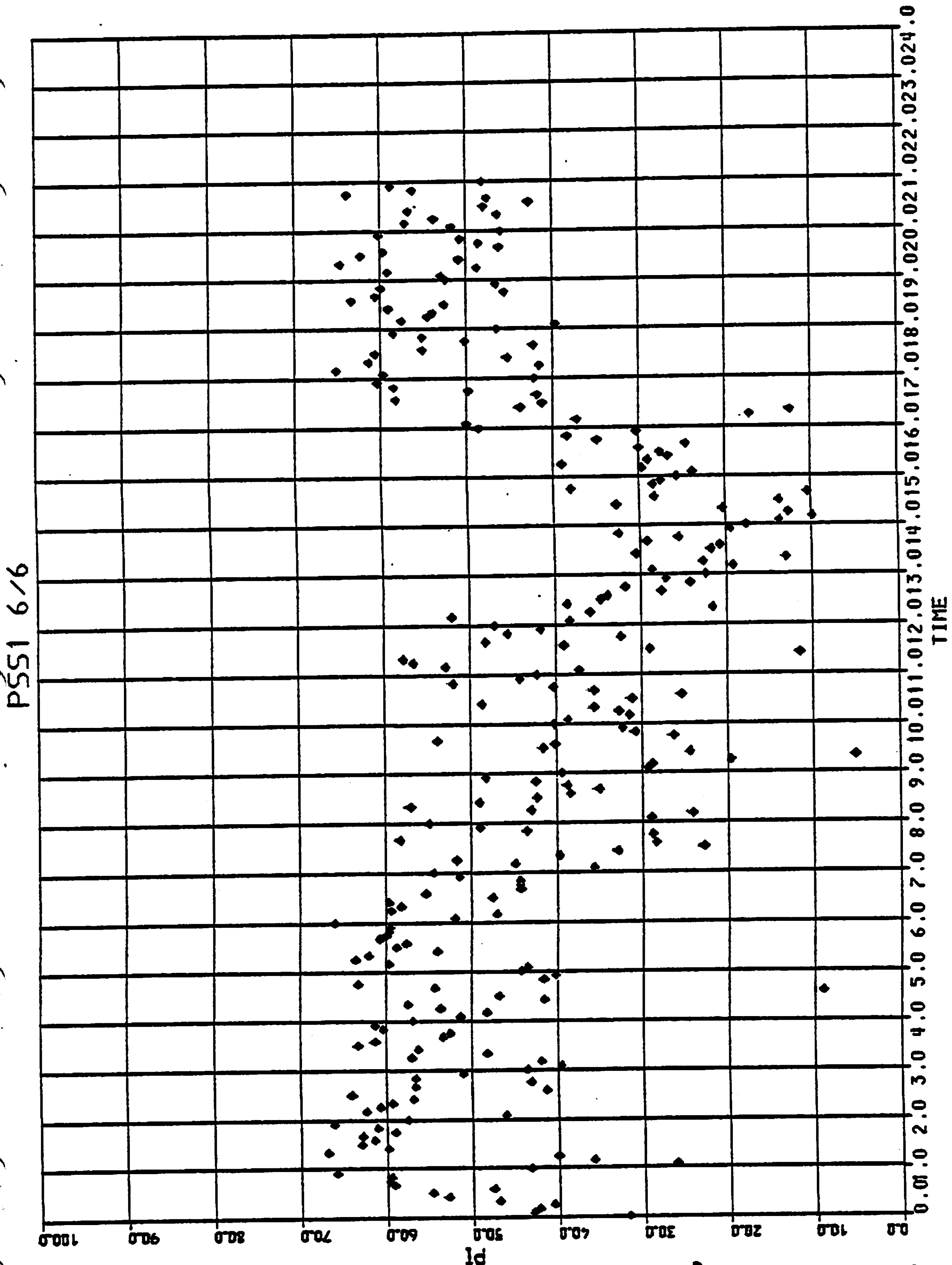
Appendix 8.7

JOB NAME=JLP***** TSS NO=01E2 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=10:55:07 PAGE NO 001

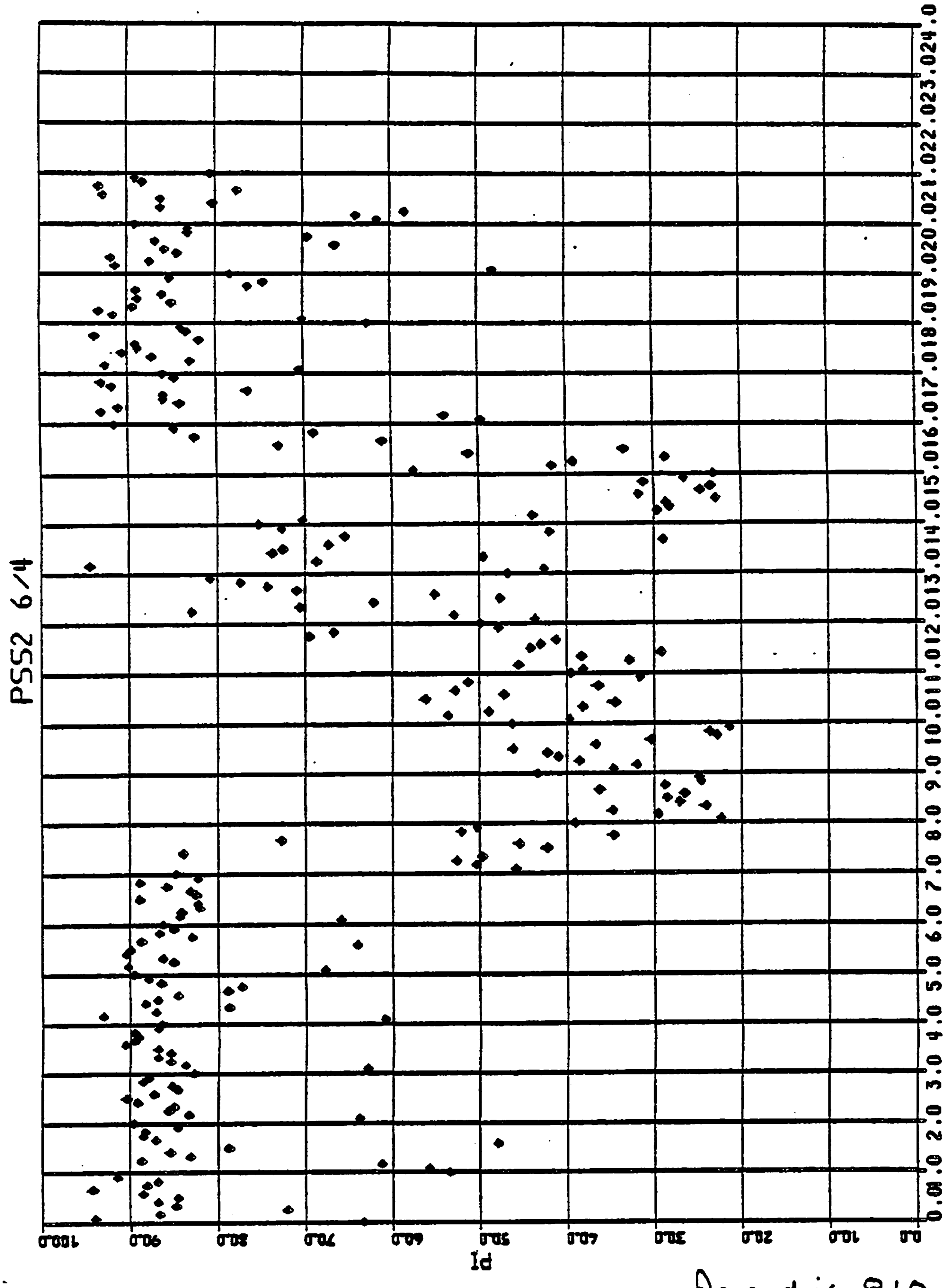


Appendix 8.8

JOB NA:E=JLP***** TSS NO=):ES BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=10:56:37 PAGE NO 001

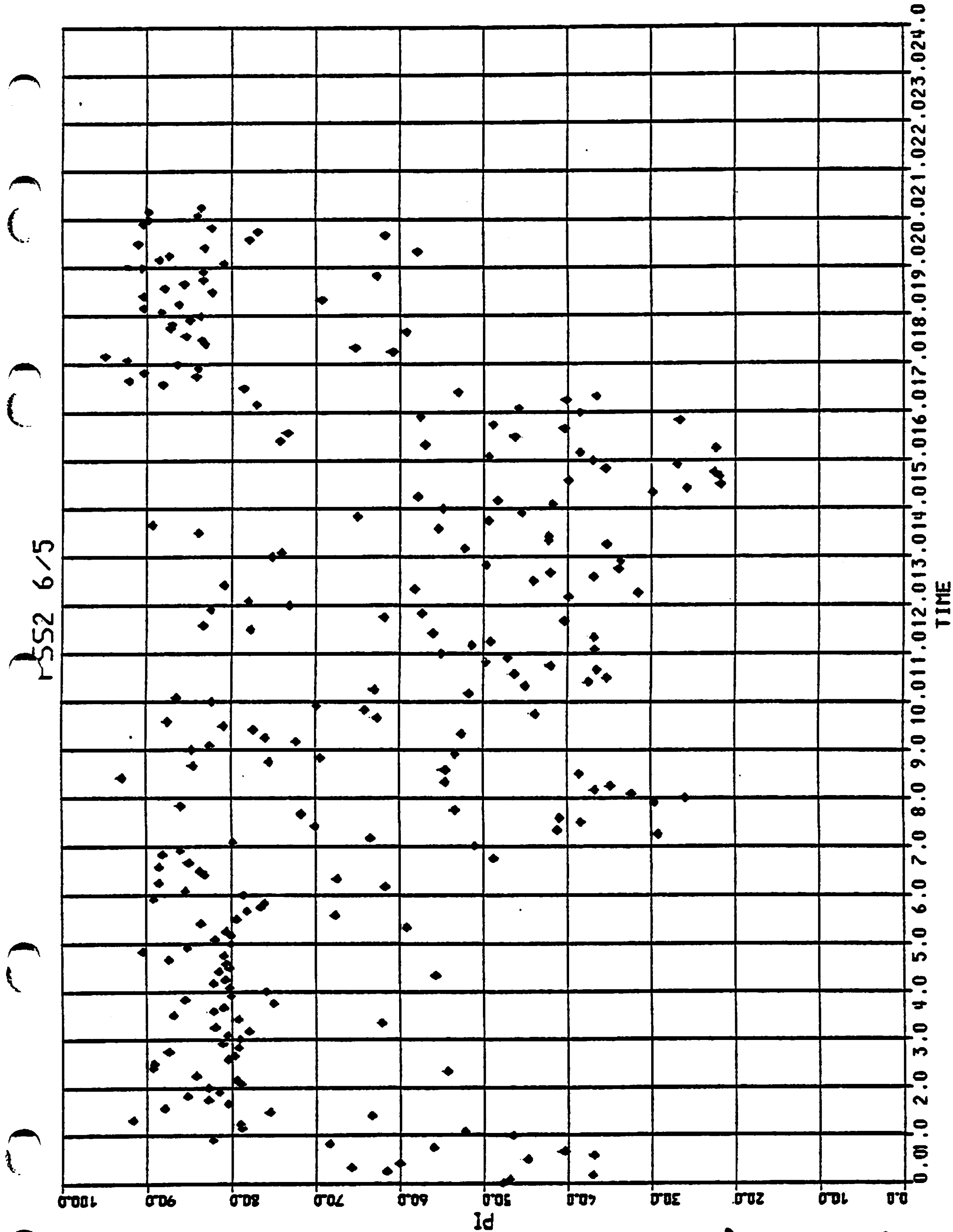

PSS1 6/6
TIME
PI
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
24.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0

Appendix 8.9



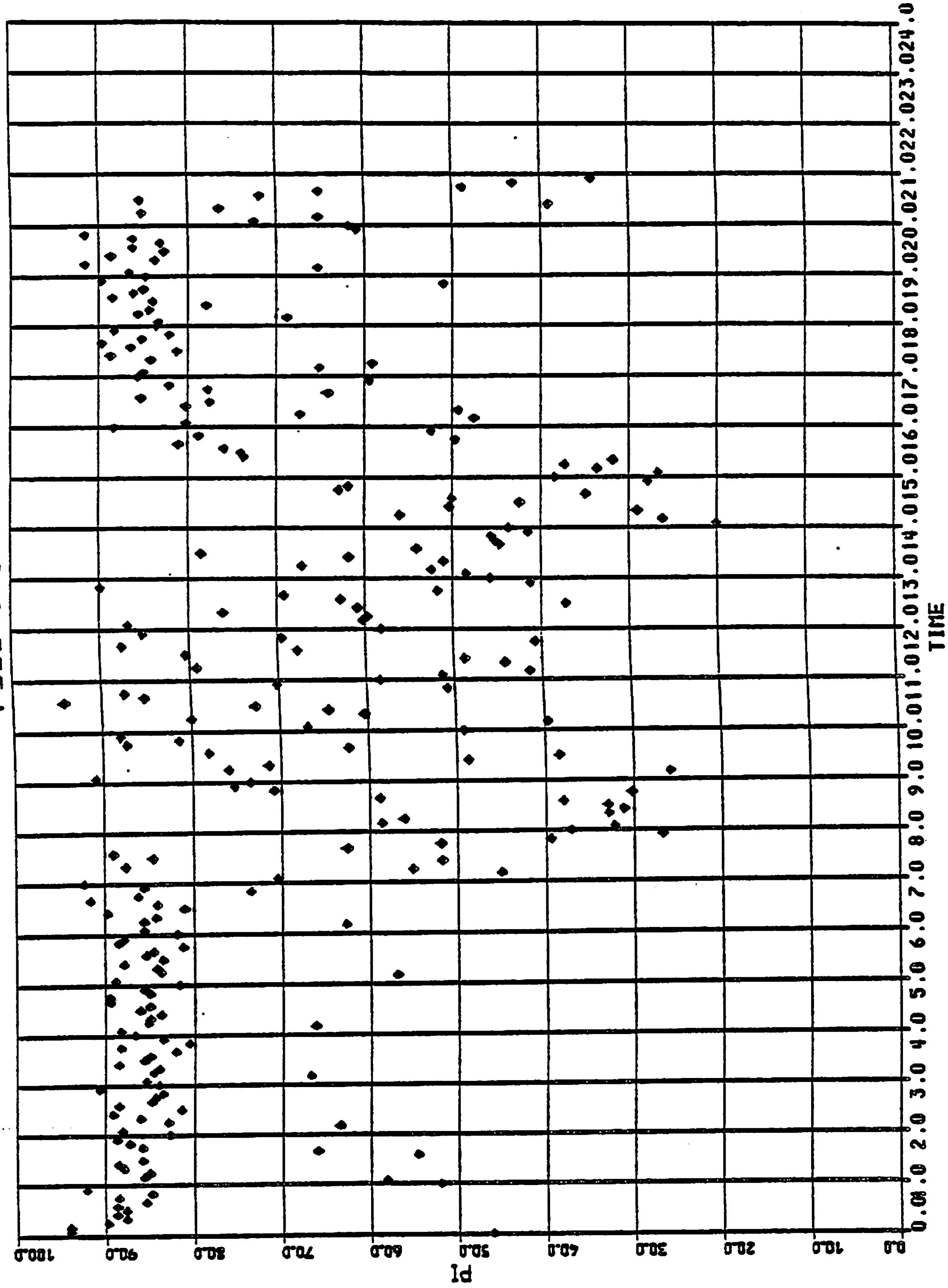
Appendix 8.10

JOB NAME=JLP***** TSS NO=01F9 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=11:04:42 PAGE NO 001



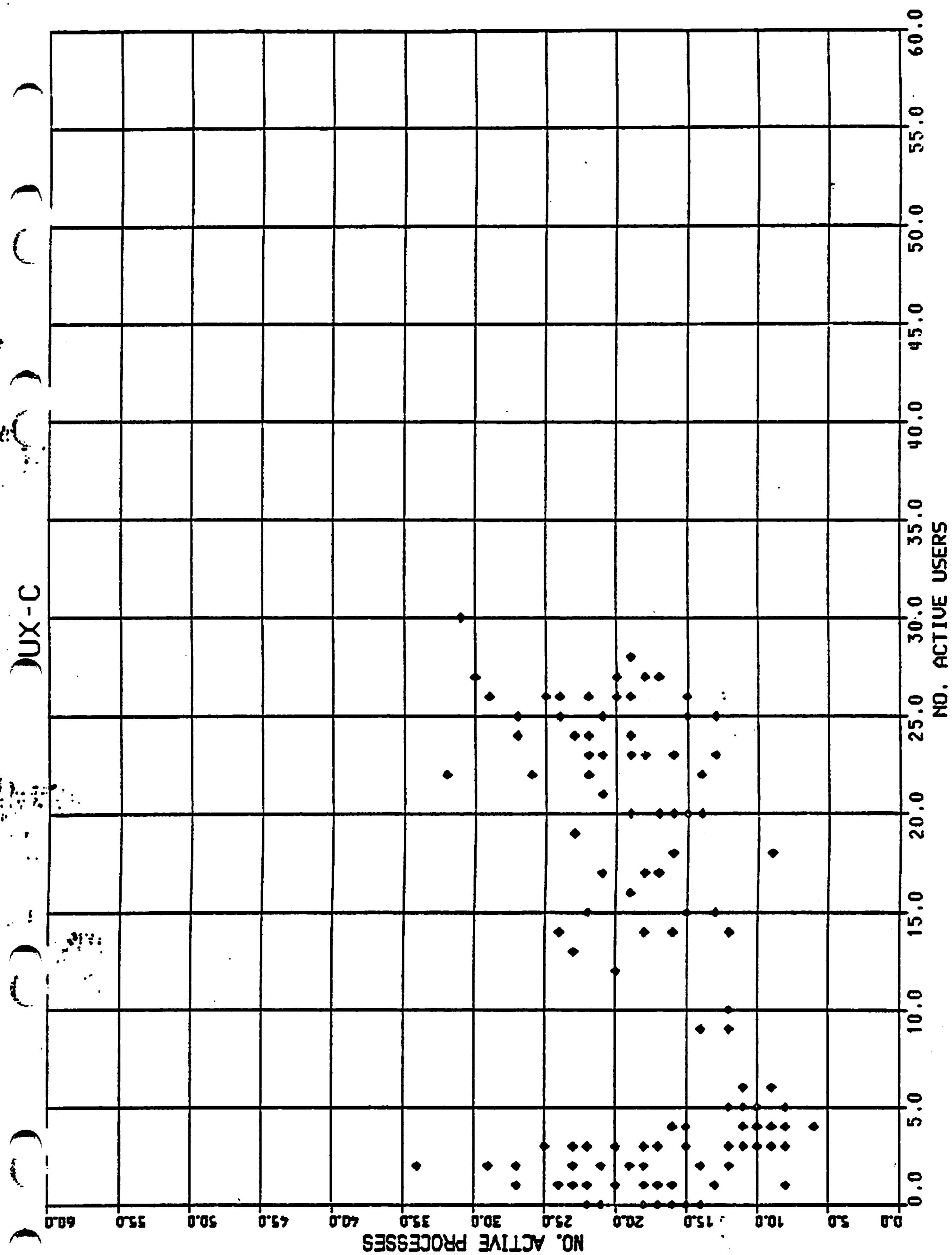
Appendix 8.11

PSS2 6/6



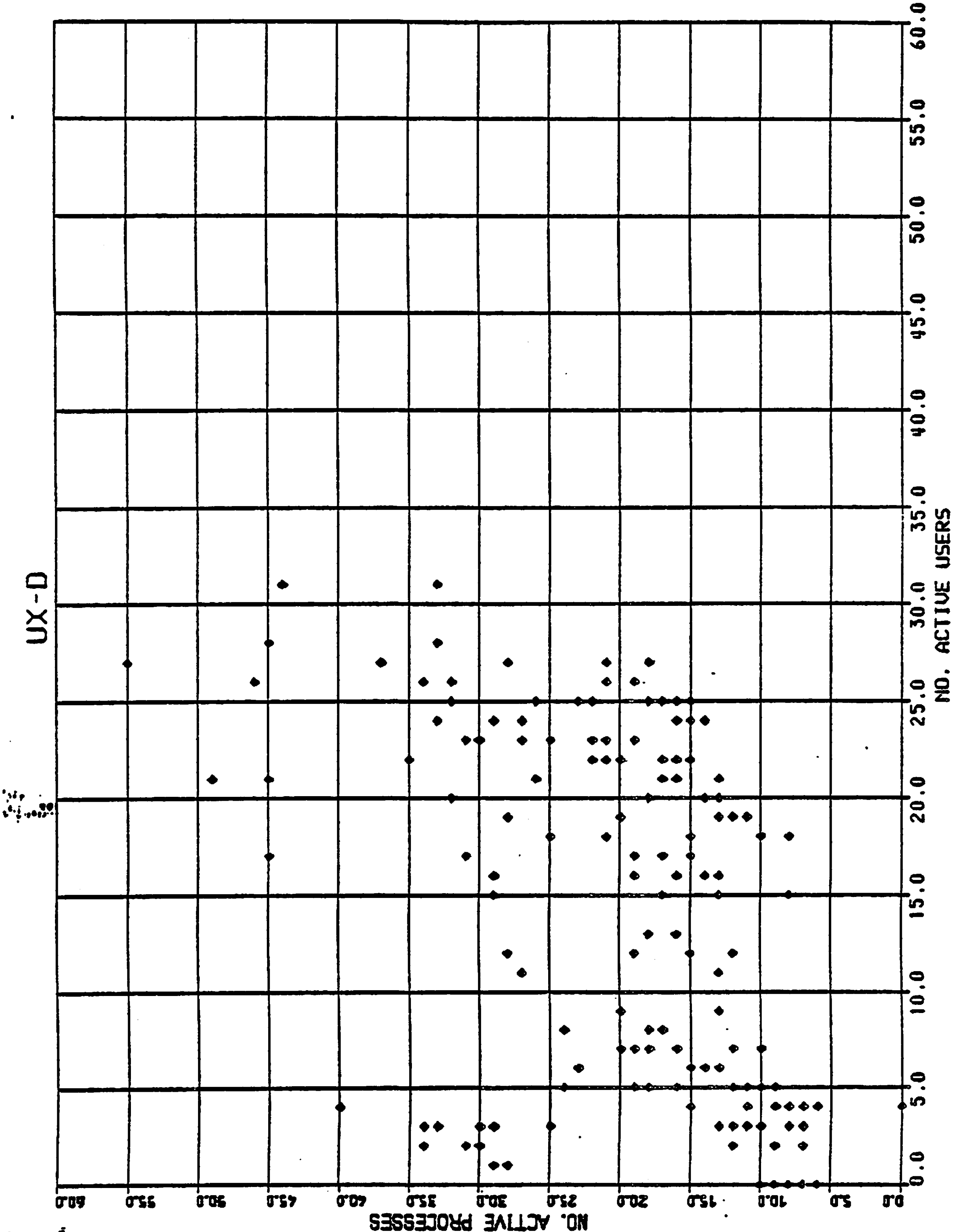
Appendix 8.12

JOB NAME=JLP***** TSS NO=01FD BIN=0360 NAME=PAPPAS
MAIL= DATE=06/08/79 TIME=11:07:49 PAGE NO 001



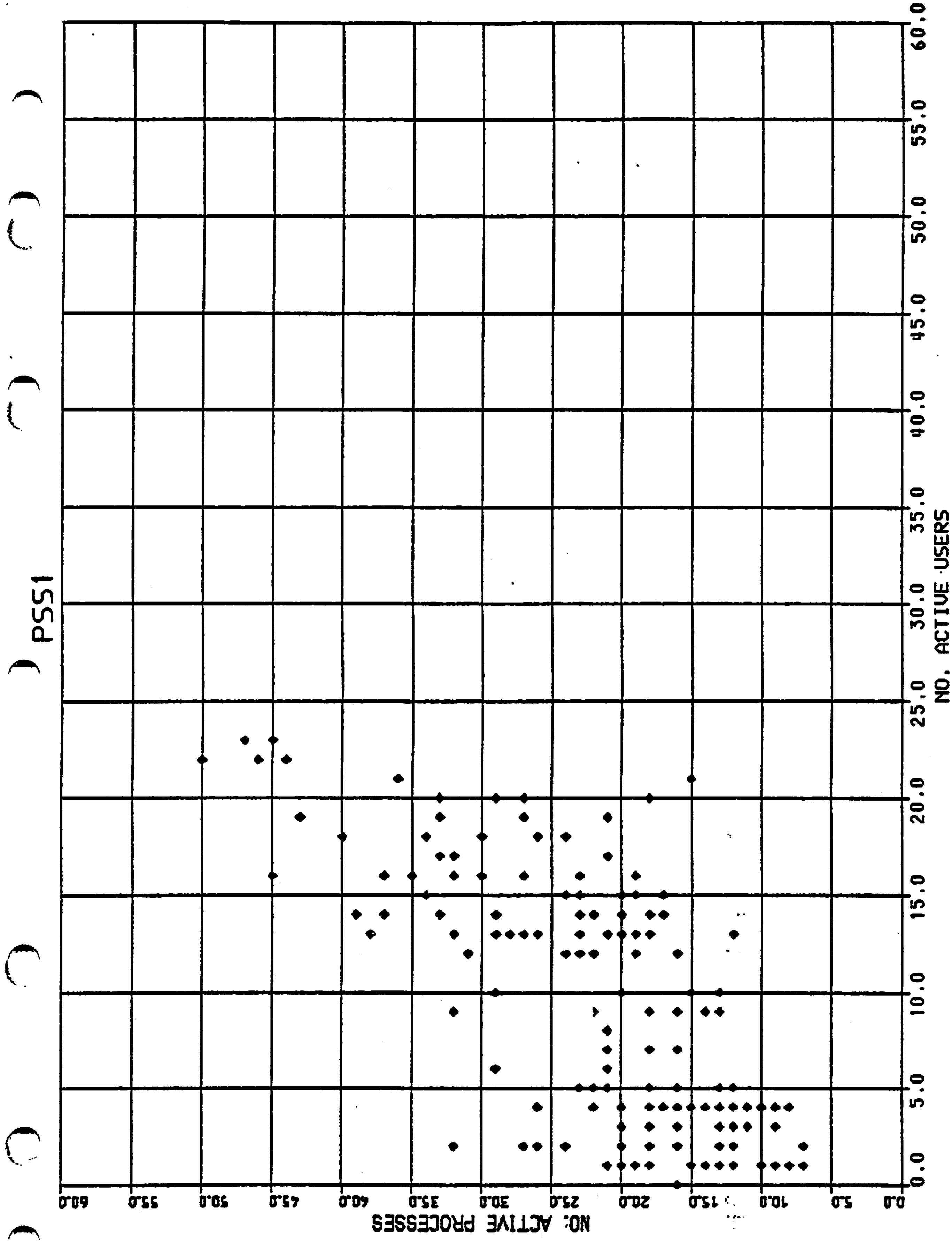
Appendix 9.1

JOB NAME=JLP××××× TSS NO=0105 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/13/79 TIME=08:55:31 PAGE NO 001



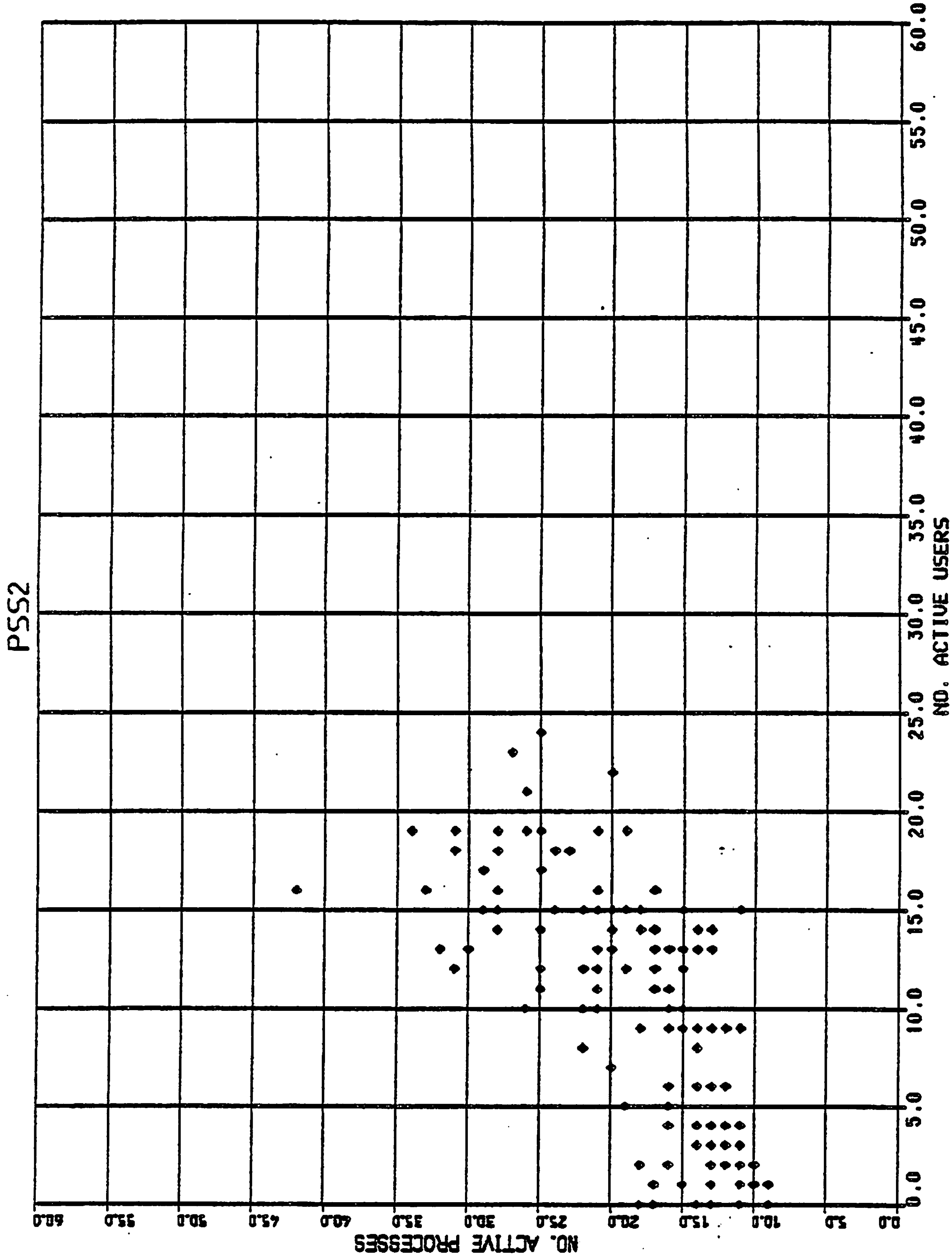
Appendix 9.2

JOB NAME=JLP×××××   TSS NO=00CD   BIN=0360   NAME=PAPPAS
MAIL=
DATE=06/13/79   TIME=08:53.38   PAGE NO 001



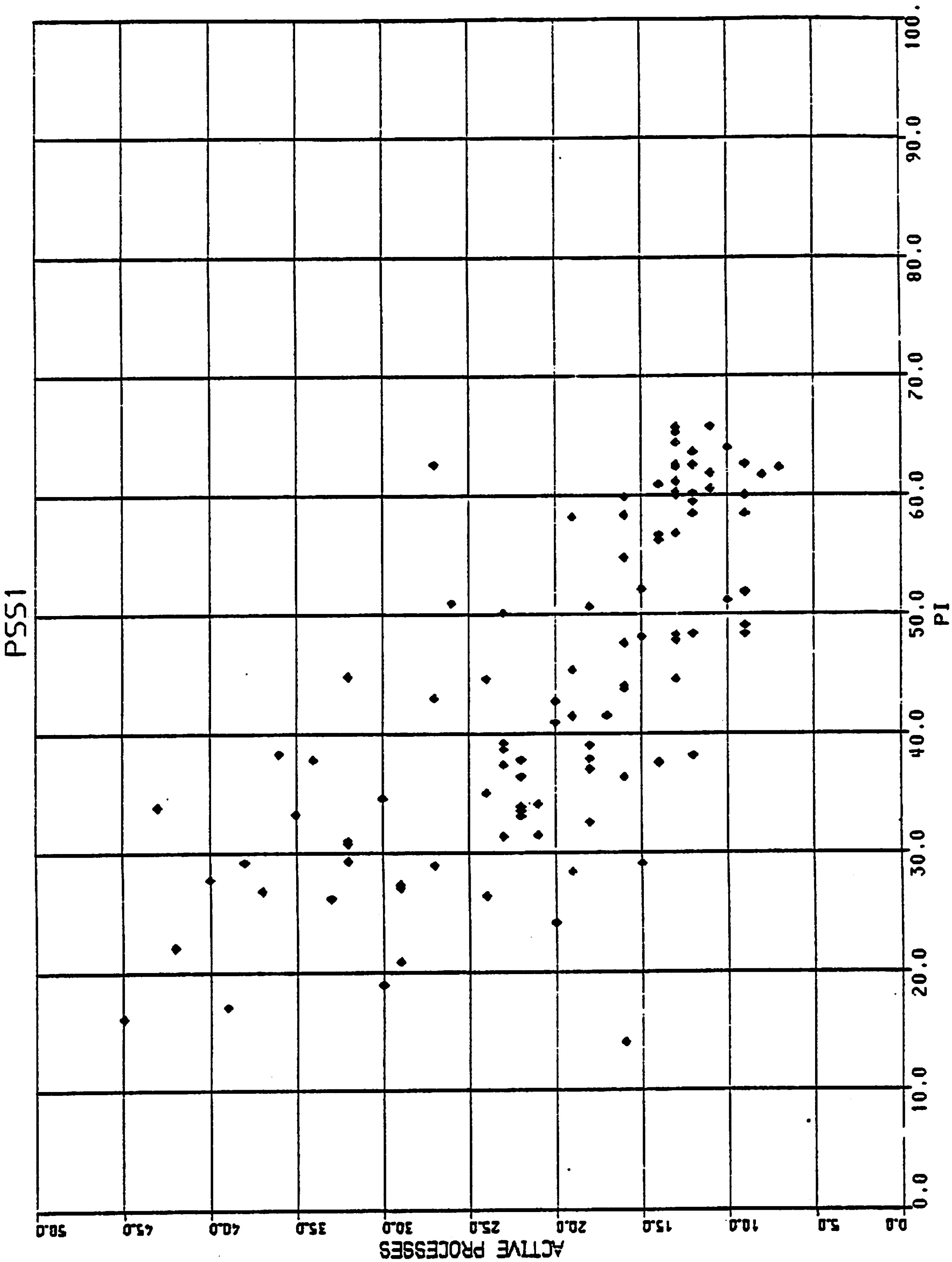
Appendix 9.3

JOB NAME=JLPxxxxx TSS NO=0392 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/13/79 TIME=15:15:58 PAGE NO 001



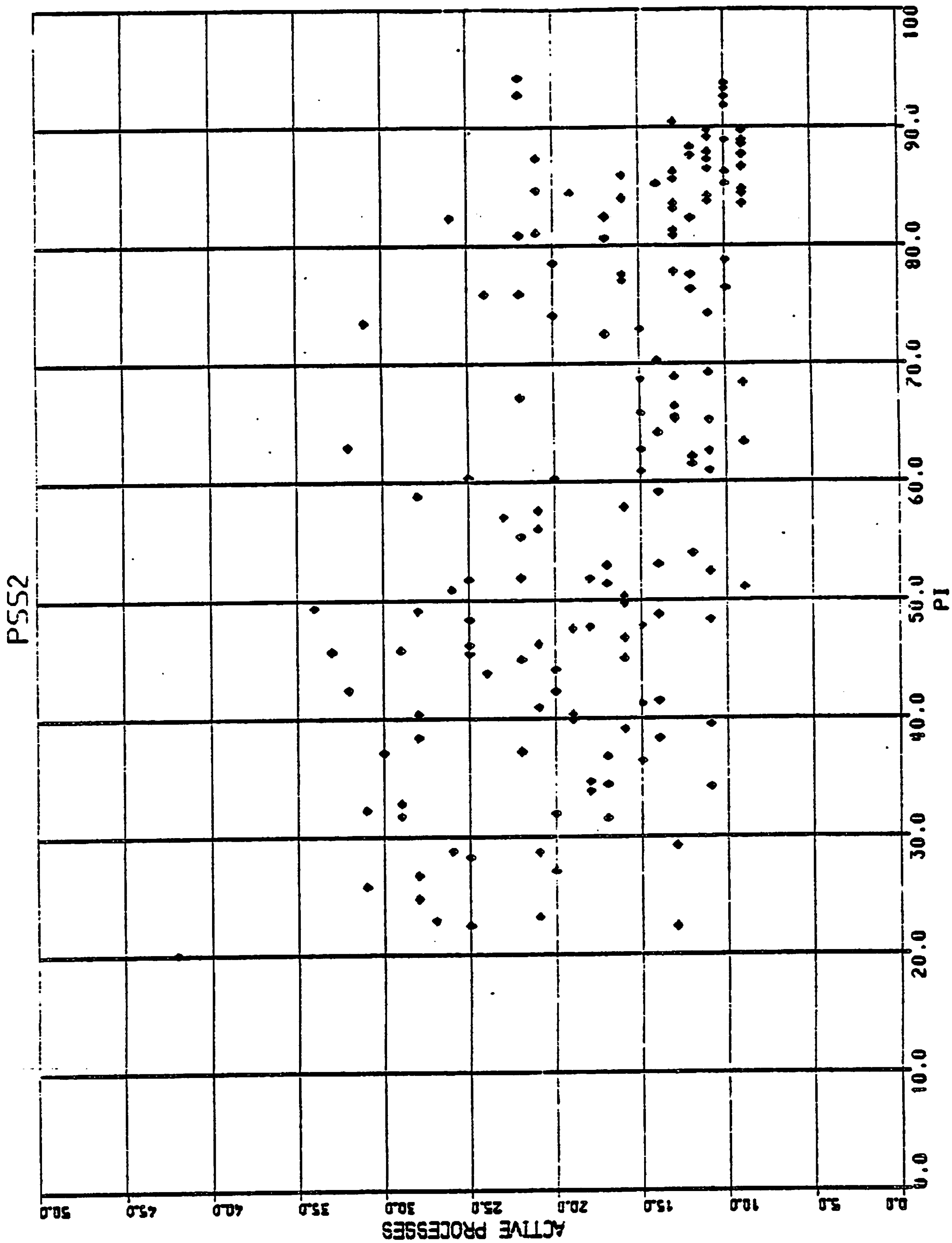
Appendix 9.4

JOB NAME=JLP****** TSS NO=0155 BIN=0360 NAME=PAPPAS
MAIL=
DATE=06/14/79 TIME=10:23:33 PAGE NO 001



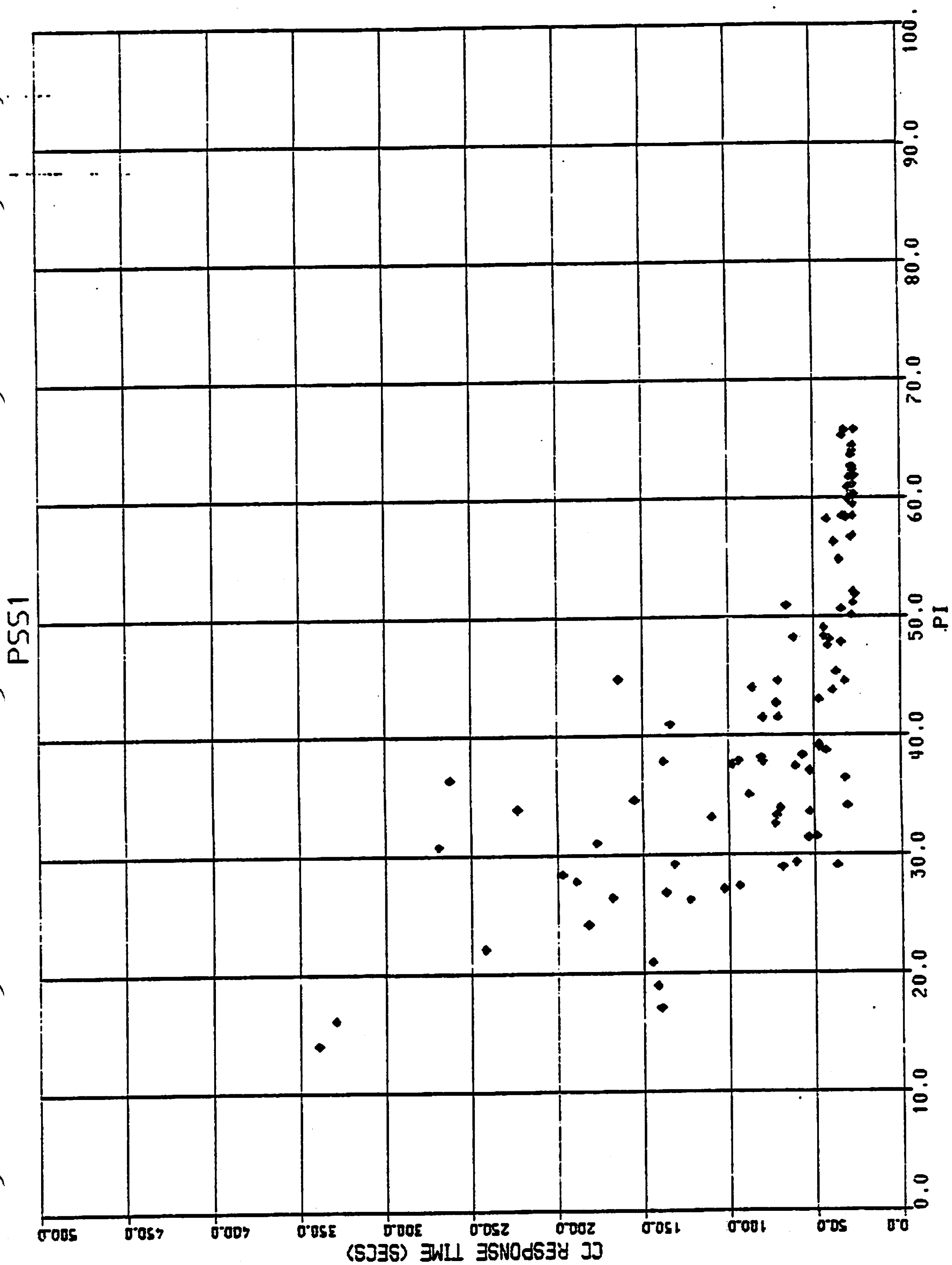
Appendix 10.1

JOB NAME=JLPxxxxx TSS NO=02BD BIN=0360 NAME=PAPPAS
DATE=06/2[illegible]/79 TIME=13:[illegible]:51 PAGE NO 001



Appendix 10.2

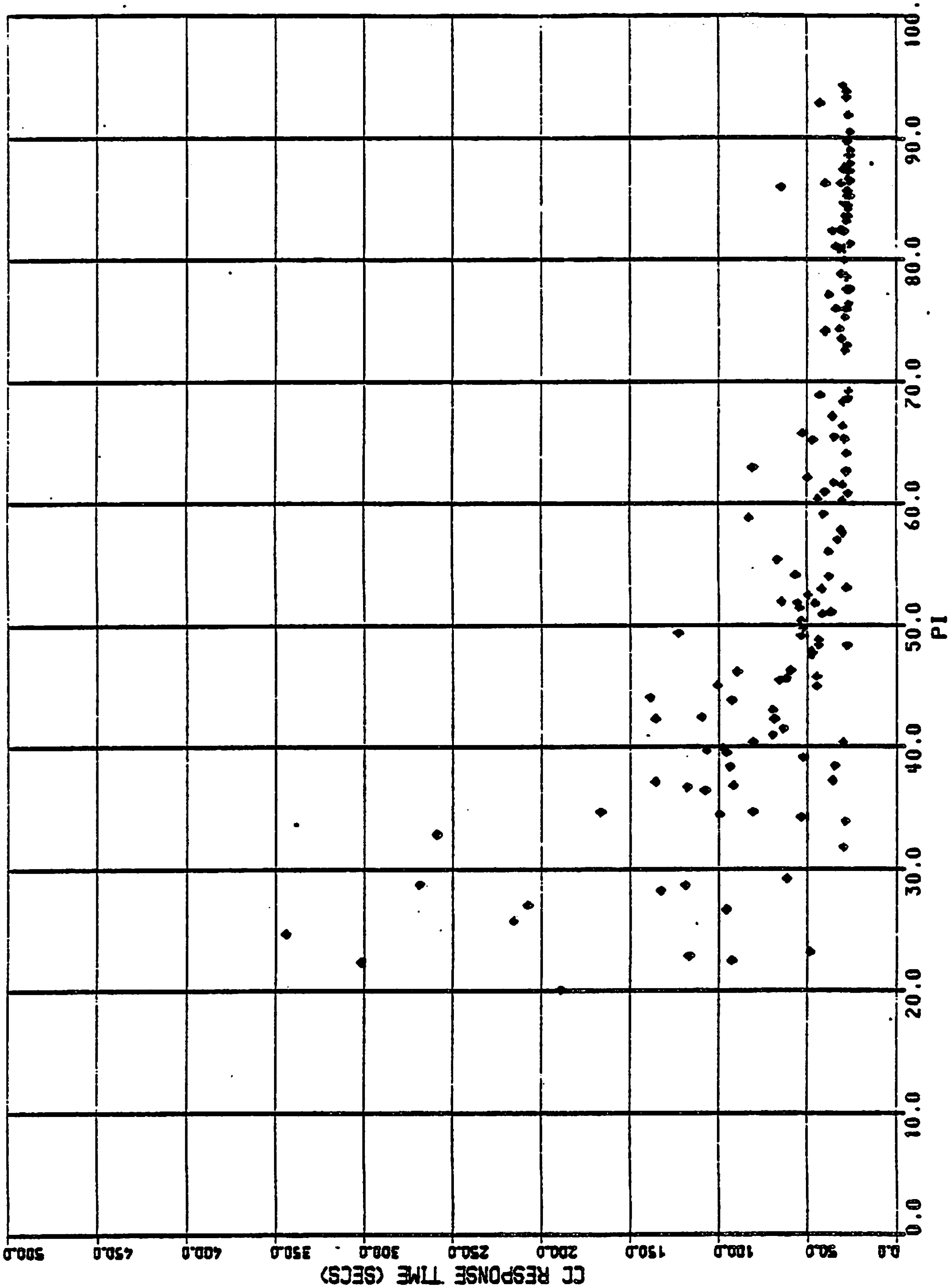
JOB NAME=JLPxxxxx TSS NO=02C1 BIN=0360 NAME=PAPPAS
MAIL= DATE=06/27/79 TIME=13:38:19 PAGE NO 001



Appendix 11.1

JOB NAME=JLP***** TSS NO=0281 BIN=0360 NAME=PAPPAS
DATE=06/27/79 TIME=13:25:16 PAGE NO 001

PSS2



Appendix 11.2

JOB NAME=JLP***** TSS NO=02A1 BIN=0360 NAME=PAPPAS

DATE=06/27/79 TIME=13:[illegible]:55 PAGE NO 001

High Runners - UNIX C, 0604

time	%CPU*	TCPU (min)	#exc. commands**
5am	usr-uni-io	daystats= 1.85	14
6am	sgs-nw-db	3bld = 4.79	50
7am	usr-cms-misc	mrstat = 1.48	4
8am	nw-uni-usr	xcrypt = 5.51	5
9am	usr-sgs-uni	nroff = 3.57	8
10am	usr-sgs-doc	rm = 4.74	473
11am	usr-nw-sgs	rm = 5.53	572
12pm	usr-nw-sgs	rm = 2.84	262
1pm	sgs-usr-cms	mrstat = 3.95	14
2pm	cms-db-usr	mrstat = 6.96	1
3pm	usr-doc-cms	nroff = 4.53	6
4pm	sgs-usr-doc	nroff = 3.65	6
5pm	usr-doc-sgs	nroff = 1.63	3
6pm	usr-misc-sgs	zat = 1.36	36

* % CPU - ranked left to right ⇒ high to low

** #exc. commands = # of commands executed by command in TCPU column

High Runners - UNIX C, 0605

time	% CPU*	TCPU	# exc commands**
5am	usr-db-io	nfind = 1.28	2
6am	db-usr-nw	dd = 1.43	1
7am	usr-misc-cms	zat = 1.4	36
8am	nw-usr-doc	nroff = 3.09	8
9am	doc-usr-nw	nroff = 4.32	12
10am	usr-doc-sgs / nw	nroff = 2.78	2
11am	sgs-doc-nw	nroff = 7.81	8
12pm	doc-usr-nw	nroff = 5.99	8
1pm	sgs-doc-usr	3bld = 4.64	50
2pm	usr-nw-sgs	nroff = 3.25	8
3pm	doc-usr-nw	nroff = 12.24	22
4pm	doc-usr-uni	nroff = 10.28	16
5pm	sgs-db-io	dd = 8.86	1
6pm	io-usr-db	epio = 6.74	4

* % CPU = ranked left to right ⇒ high to low

** # exc commands = # of commands executed by command in TCPU column

Appendix 12.2

High Runners - UNIX C, 0606

time	%CPU*	TCPU	#exc command**
5am	usr-db-uni	dd=1.98	3
6am	usr-uni-sgs	ectgen=1.79	1
7am	usr-doc-misc	nroff=1.43	1
8am	usr-nw-doc	nroff=2.99	3
9am	sgs-doc-usr	nroff=5.42	7
10am	io-sgs-nw	cpio=7.48	2
11am	sgs-usr-doc	nroff=3.19	9
12pm	sgs-nw-usr	3bld=2.62	18
1pm	doc-usr-sgs	nroff=4.14	13
2pm	sgs-doc-usr	nroff=5.31	15
3pm	usr-sgs-doc	bsh=1.68	140
4pm	sgs-usr-doc	3bcO=3.61	17
5pm	db-io-nw	dd=12.6	3
6pm	usr-misc-sgs	sat=1.35	36

* %CPU - ranked left to right → high to low
** #exc commands = # of commands executed by command in TCPU column

Appendix 12.3

High Runners - UNIX D,0604

time	%CPU*	TCPU	#exc commands**
5am	io-db-usr	dd = 3.61	2
6am	nw-usr-misc	eat = 1.36	36
7am	usr-misc-doc	eat = 1.34	36
8am	sgs-doc-usr	nroff = 4.56	8
9am	sgs-usr-doc	bsh = 1.65	149
10am	sgs-usr-doc	nroff = 3.47	6
11am	doc-sgs-nw	nroff = 6.45	12
12pm	sgs-usr-sys	cpp = 3.4	28
1pm	sgs-doc-usr	3bdis = 3.55	4
2pm	sgs-usr-doc	cpp = 2.91	34
3pm	sgs-usr-doc	nroff = 2.3	8
4pm	sgs-doc-usr	nroff = 3.73	12
5pm	io-sgs-usr	gsi = 17.89	2
6pm	usr-sgs-misc	eat = 1.36	36

* %CPU - ranked left to right → high to low
** #exc commands = # of commands executed by command in TCPU column

Appendix 12.4

High Runners - UNIX D, 0605

time	%CPU*	TCPU	#exc commands**
5am	db-doc-usr	dd=4.2	2
6am	nw-usr-misc	sat=1.36	36
7am	doc-usr-misc	nroff=6.02	11
8am	doc-usr-nw	nroff=10.48	9
9am	sgs-doc-usr	nroff=6.55	7
10am	sgs-uni-usr	c=12.18	2
11am	sgs-usr-doc	cpp=4.53	37
12pm	sgs-usr-nw	3bld=4.89	30
1pm	usr-doc-sgs	nroff=4.72	14
2pm	sgs-usr-doc	nroff=3.4	9
3pm	sgs-usr-doc	nroff=5.64	8
4pm	sgs-doc-usr	nroff=7.69	9
5pm	io-db-sgs	cpio=8.61	1
6pm	sgs-usr-misc	sat=1.44	36

* %CPU - ranked left to right → high to low
** #exc command = # of commands executed by command in TCPU column

Appendix 12.5

High Runners - UNIX D.0606

time	%CPU*	TCPU	#exc commands**
5am	usr-db-io	dd=1.98	3
6am	doc-nw-usr	nroff= 2.5	2
7am	sgs-usr-misc	cpp=1.73	18
8am	sgs-nw-usr	cpp = 4.56	47
9am	doc-usr-sgs	nroff = 6.95	16
10am	doc-usr-sgs	nroff =15.39	13
11am	usr-nw-doc	nroff = 3.22	6
12pm	sgs-usr-doc	bsh =1.15	86
1pm	sgs-usr-doc	nroff = 3.83	6
2pm	usr-sgs-nw	ftp = 3.75	9
3pm	sgs-cms-doc	get = 4.47	3
4pm	sgs-usr-doc	3bld = 3.31	19
5pm	doc-usr-sgs	nroff = 12.87	3
6pm	usr-nw-io	nfind = 12.98	2

*%CPU - ranked left to right → high to low
** #exc commands - # of commands executed by command in TCPU column

Appendix 12.6

High Runners - IHPSI, 0604

time	%CPU*	TCPU	#exc commands**
5am	nw-usr-misc	bln.psl. = 4.3	3
6am	nw-usr-misc	bln.psl. = 1.99	1
7am	nw-usr-doc	bln.psl. = 1.99	1
8am	nw-usr-sgs	bln.psl. = 1.99	1
9am	db-usr-uni	sort = 9.1	2
10am	uni-usr-nw	a.out = 8.55	3
11am	usr-nw-cms	bln.psl. = 3.31	1
12pm	nw-uni-usr	vc10 = 3.65	2
1pm	db-nw-usr	sort = 10.72	4
2pm	usr-db-nw	sort = 8.15	5
3pm	usr-cms-nw	sort = 3.52	18
4pm	nw-usr-cms	bln.psl. = 3.78	1
5pm	usr-nw-misc	chess3 = 8.19	1
6pm	usr-sgs-misc	chess3 = 2.36	1

* % CPU - ranked left to right → high to low
** #exc commands - # of commands executed by command in TCPU column

Appendix 12.7

High Runners - IHPSI, 0605

time	%CPU*	TCPU	#exc commands**
5am	nw-usr-misc	bln.psl. = 2.49	1
6am	nw-usr-misc	bln.psl. = 2.49	1
7am	nw-usr-misc	bln.psl. = 2.49	1
8am	doc-usr-nw	nroff = 3.52	7
9am	db-usr-doc	sort = 8.19	12
10am	usr-cms-nw	bsh = 4.33	139
11am	usr-nw-io	bln.psl. = 4.79	1
12pm	usr-nw-db	bln.psl. = 4.79	1
1pm	nw-usr-db	bln.psl. = 4.79	1
2pm	usr-cms-nw	grep = 4.12	414
3pm	usr-cms-nw	m2edit = 6.21	4
4pm	usr-nw-cms	bln.psl. = 4.93	2
5pm	usr-nw-misc	sat = 1.57	36
6pm	usr-misc-nw	sat = 1.53	36

* %CPU - ranked left to right → high to low

** #exc commands - # of commands executed by command in TCPU column

Appendix 12.8

High Runners - IHPS1, 0606

time	% CPU*	TCPU	#exc commands**
5am	usr-nw-cms	bln.psl. = 2.95	1
6am	nw-usr-misc	bln.psl = 2.95	1
7am	nw-usr-misc	bln.psl. = 2.95	1
8am	nw-usr-doc	nroff = 4.11	11
9am	usr-nw-cms	bln.psl. = 4.42	1
10am	usr-nw-doc	bsh = 4.31	165
11am	usr-nw-db	bsh = 4.39	186
12pm	usr-nw-cms	bln.psl. = 4.04	2
1pm	usr-nw-db	bln.psl. = 4.05	1
2pm	usr-nw-cms	bsh = 4.59	182
3pm	usr-cms-nw	nroff = 4.23	22
4pm	usr-nw-doc	nroff = 5.74	13
5pm	nw-usr-sys	bln.psl. = 5.9	1
6pm	nw-usr-misc	bln.psl. = 5.89	1

* % CPU - ranked left to right → high to low

** #exc commands - # of commands executed by command in TCPU column

Appendix 12.9

High Runners - IHPS2, 0604

time	%CPU*	TCPU	#exc. commands**
5am	misc-usr-nw	eat = 1.36	36
6am	misc-usr-nw	eat = 1.36	36
7am	misc-usr-nw	eat = 1.36	36
8am	sgs-usr-cms	nroff = 1.58	6
9am	doc-sgs-cms	nroff = 14.43	11
10am	sgs-doc-usr	nm801 = 6.59	1
11am	sgs-usr-nw	nm801 = 6.59	1
12pm	doc-sgs-usr	nm801 = 3.48	1
1pm	usr-doc-io	troff = 3.81	1
2pm	usr-sgs-cms	nroff = 1.94	17
3pm	doc-sgs-usr	troff = 10.14	1
4pm	doc-usr-cms	troff = 3.31	1
5pm	sgs-misc-usr	eat = 1.38	36
6pm	sgs-misc-usr	eat = 1.36	36

* %CPU - ranked left to right → high to low
** #exc commands - # of commands executed by command in TCPU column

Appendix 12.10

High Runners - IHPS2, 0605

time	%CPU*	TCPU	#exc commands**
5am	misc-usr-nw	eat=1.34	36
6am	misc-usr-nw	eat=1.3	36
7am	doc-usr-cms	nroff=4.92	12
8am	doc-cms-usr	nroff=13.94	7
9am	usr-sgs-cms	bsh=1.41	126
10am	usr-cms-sgs	a.out=1.91	21
11am	uni-usr-sgs	a.out=42.03	20
12pm	usr-uni-sgs	a.out=3.09	7
1pm	sgs-usr-ab	bsh=1.77	172
2pm	cms-usr-sgs	grt=5.45	157
3pm	cms-doc-sgs	nroff=6.64	38
4pm	uni-usr-doc	start.c=8.15	1
5pm	doc-usr-misc	troff=3.43	1
6pm	sgs=usr-misc	eat=1.4	36

* %CPU - ranked left to right → high to low
** #exc commands - # of commands executed by command in TCPU column

Appendix 12.11

High Runners - IHPS2, 0606

time	%CPU*	TCPU	#exc commands**
5am	misc-usr-nw	sat=1.41	36
6am	misc-usr-nw	sat=1.38	36
7am	misc-nw-usr	sat=1.38	36
8am	nw-sgs-usr	haspqer=2.7	35
9am	usr-cms-sgs	grt=2.55	100
10am	usr-sgs-uni	bsh=1.42	107
11am	usr-doc-sgs	bsh=1.63	111
12pm	usr-doc-misc	nroff=2.1	6
1pm	usr-doc-io	s=2.34	35
2pm	usr-nw-io	diskusag=2.85	6
3pm	db-usr-cms	sgrep=12.03	8
4pm	usr-doc-nw	bsh=2.34	149
5pm	usr-nw-sgs	sat=1.46	36
6pm	sgs-misc-usr	sat=1.38	36

* %CPU - ranked left to right → high to low
** #exc commands = # of commands executed by command in TCPU column

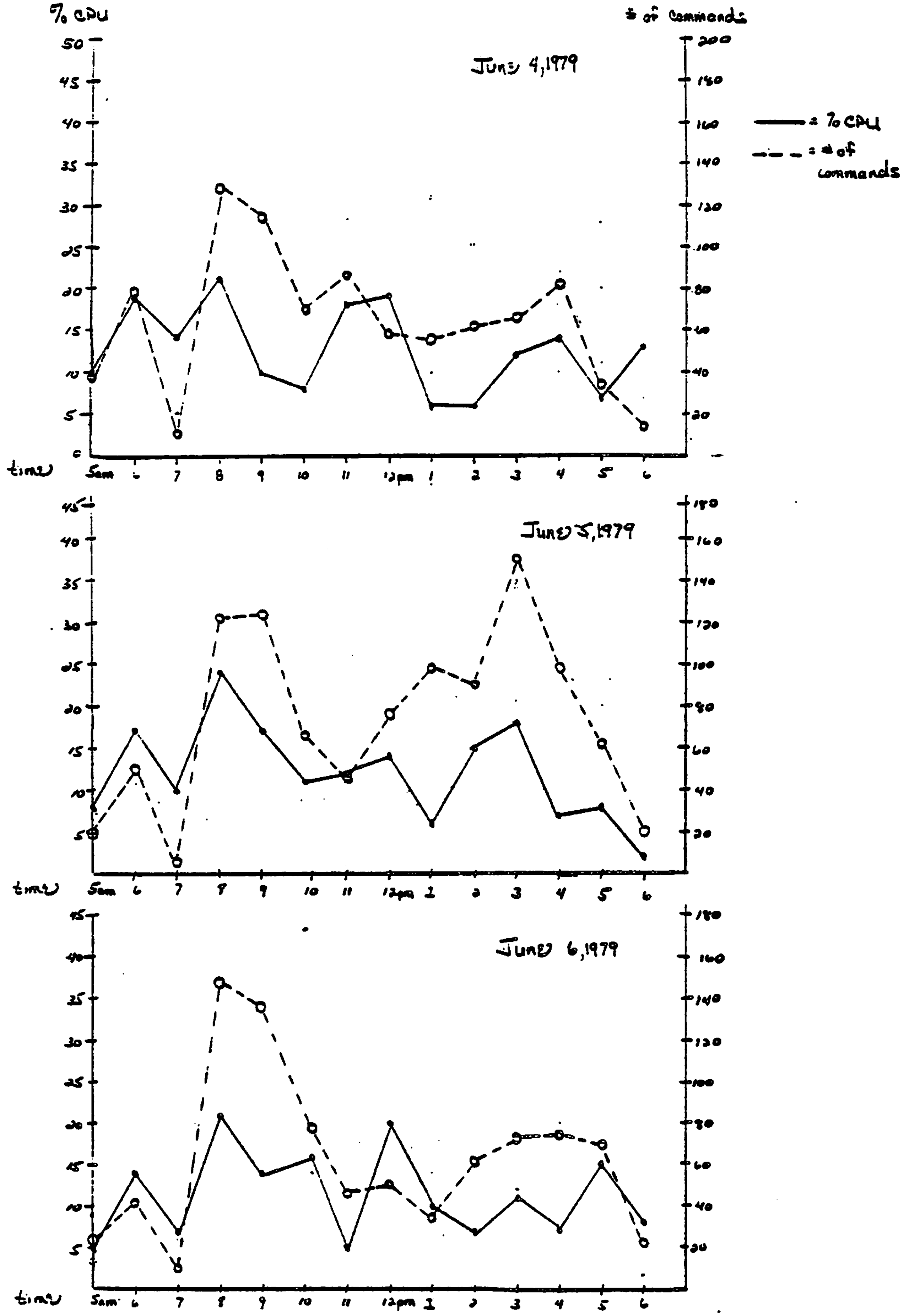
Appendix 12.12

High Runners Summarization

	UNIX C #1	UNIX C #2	UNIX C #3	UNIX D #1	UNIX D #2	UNIX D #3	IHPS1 #1	IHPS1 #2	IHPS1 #3	IHPS2 #1	IHPS2 #2	IHPS2 #3
db	2	4	2	1	3	-	3	1	4	1	-	1
doc	5	8	7	6	8	13	1	-	5	7	7	1
io	2	1	3	3	-	2	-	-	1	-	-	3
nw	2	6	8	2	4	4	15	15	9	1	3	9
sgs	11	4	8	22	4	5	-	1	2	8	7	8
usr	19	12	6	7	21	11	22	18	2	13	18	9
sys	-	-	-	-	-	1	-	-	-	-	-	-
misc	-	3	2	-	1	6	-	1	11	8	3	3
uni	-	3	3	-	1	-	1	1	1	2	1	1
cms	1	1	3	-	1	-	-	5	7	2	3	7

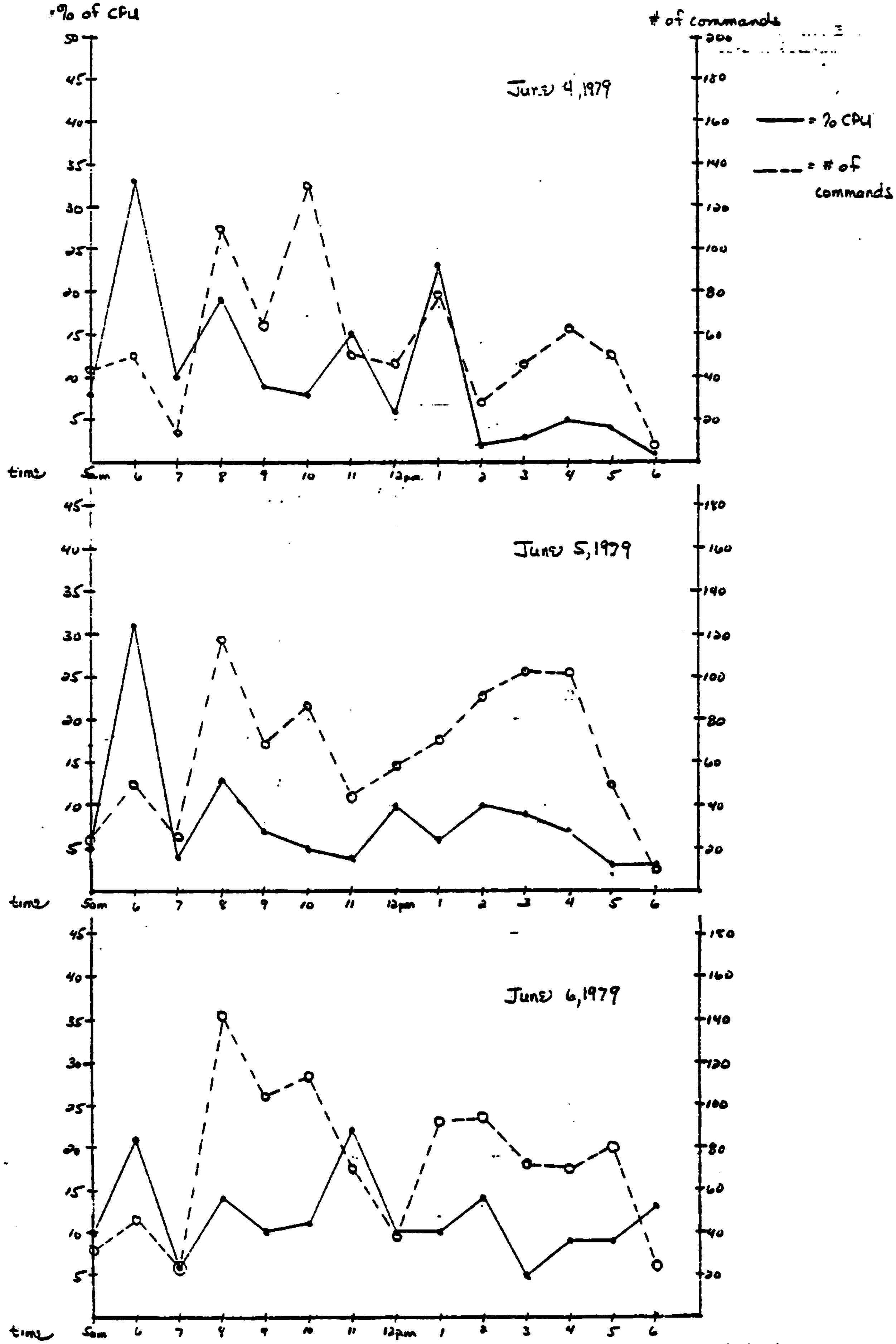
	UNIX C	UNIX D	IHPS1	IHPS2
db	16	9	15	4
doc	38	47	8	36
io	8	5	1	3
nw	26	18	84	18
sgs	49	79	4	46
usr	87	74	104	84
sys	-	1	-	-
misc	8	8	13	33
uni	9	2	6	9
cms	8	2	17	19

Appendix 13



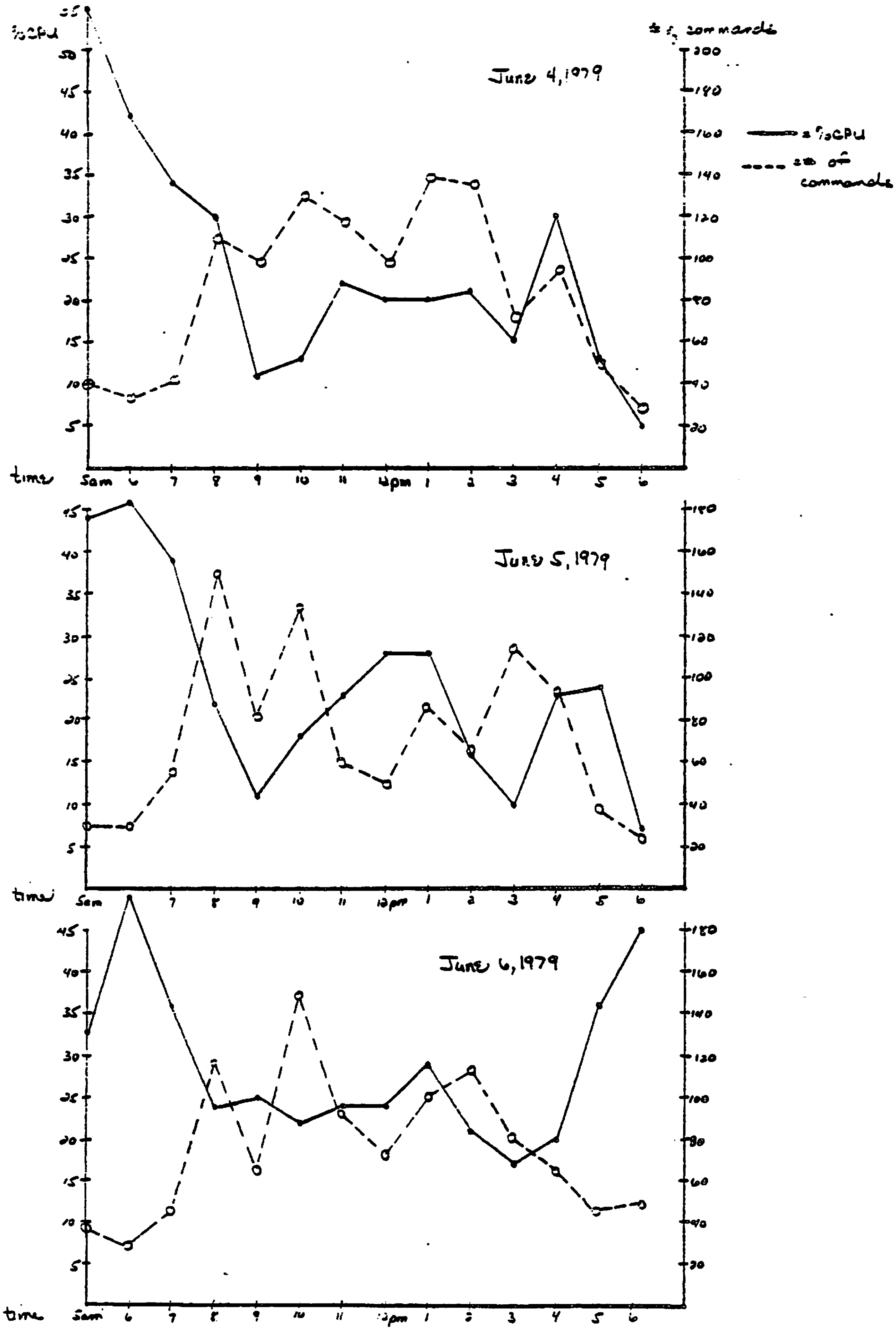
UNIX C - NETWORKING

Appendix 14.1

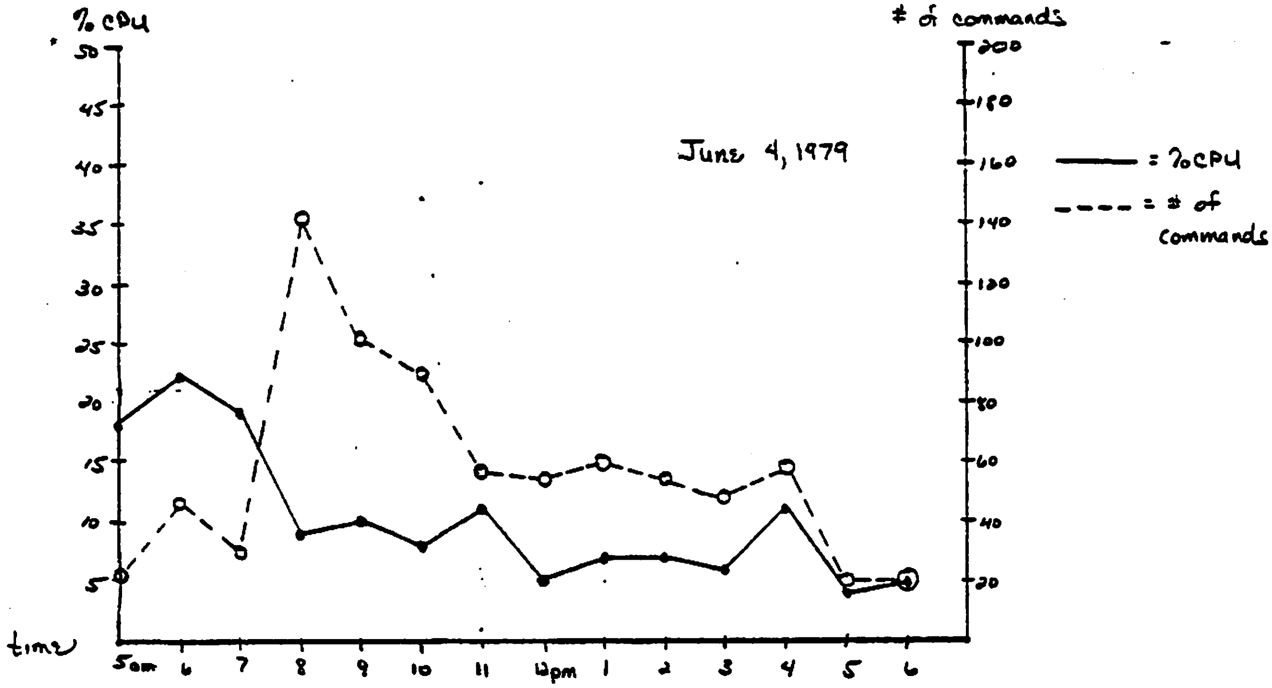


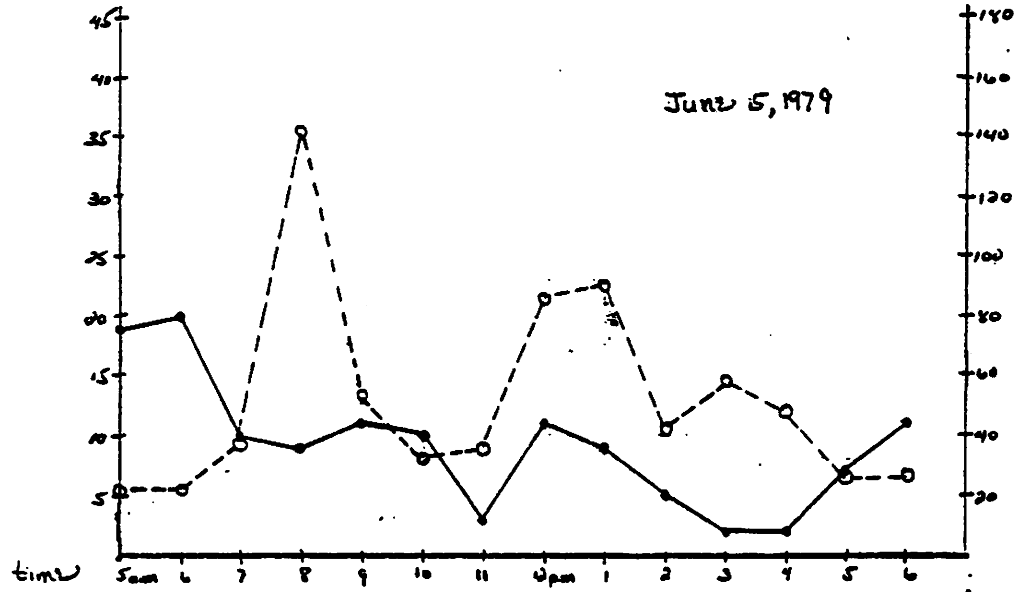
UNIX D- NETWORKING

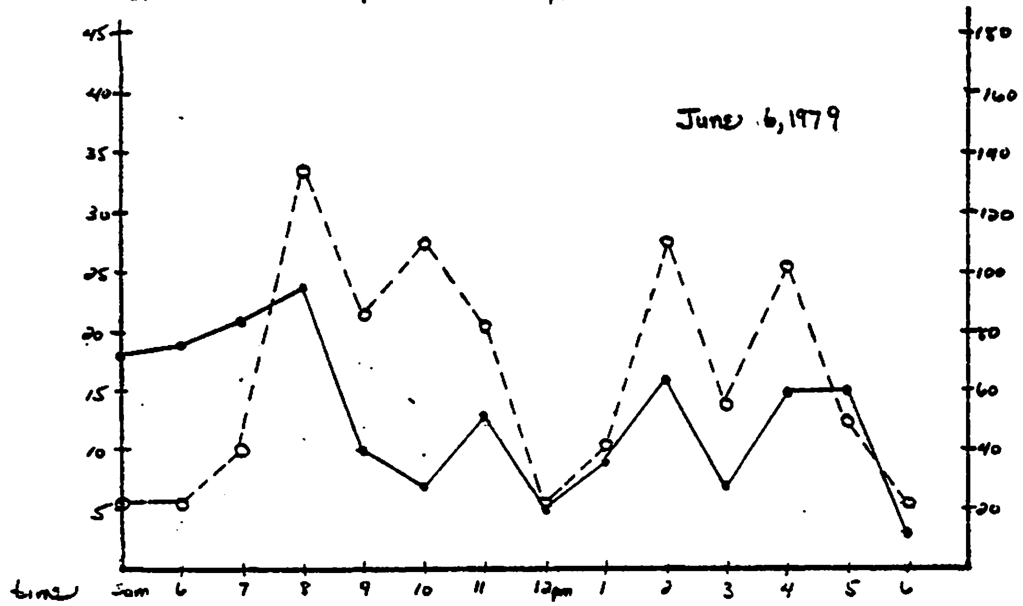
Appendix 14.2



Appendix 14.3 IHPSI - Networking

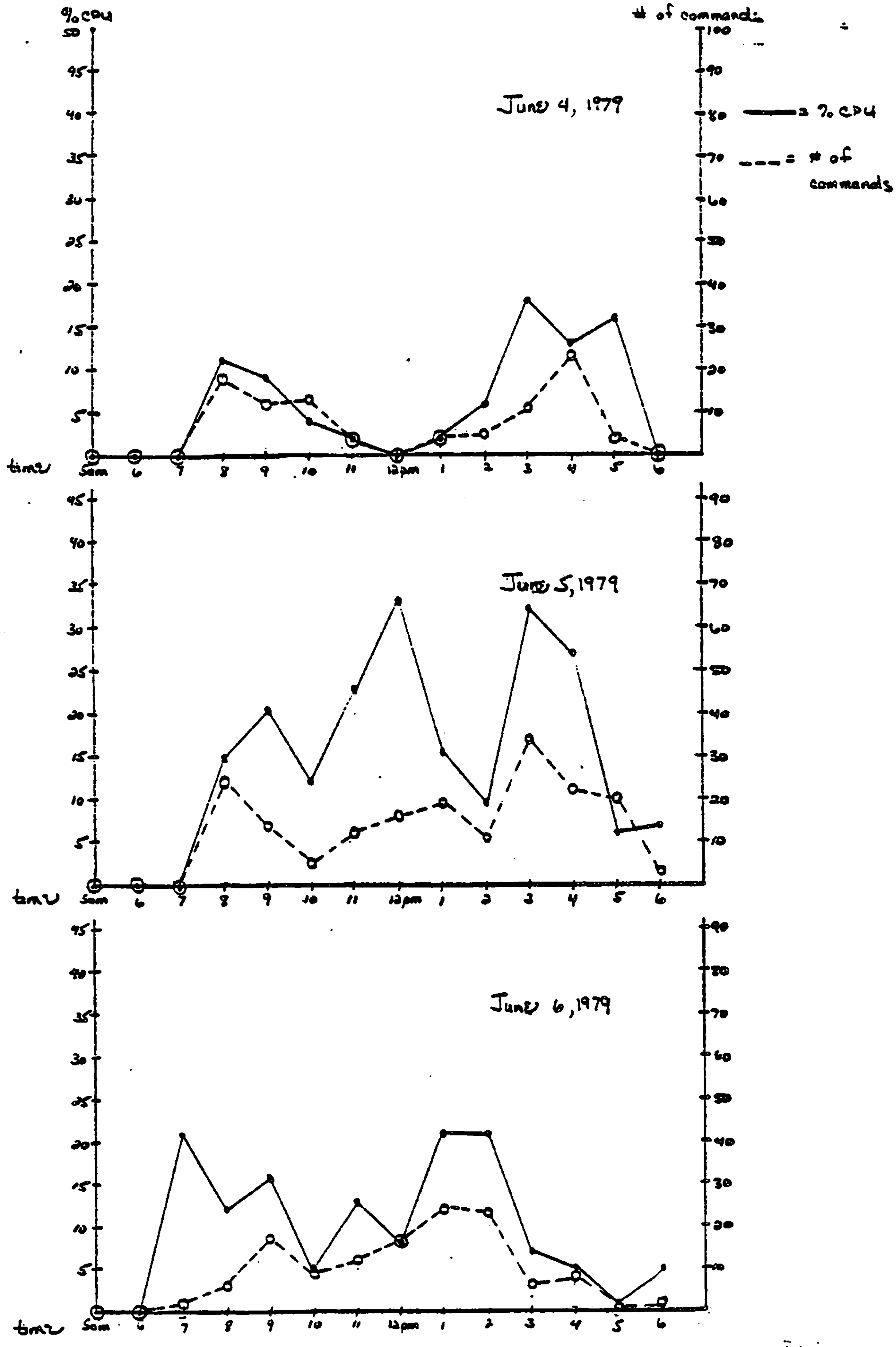






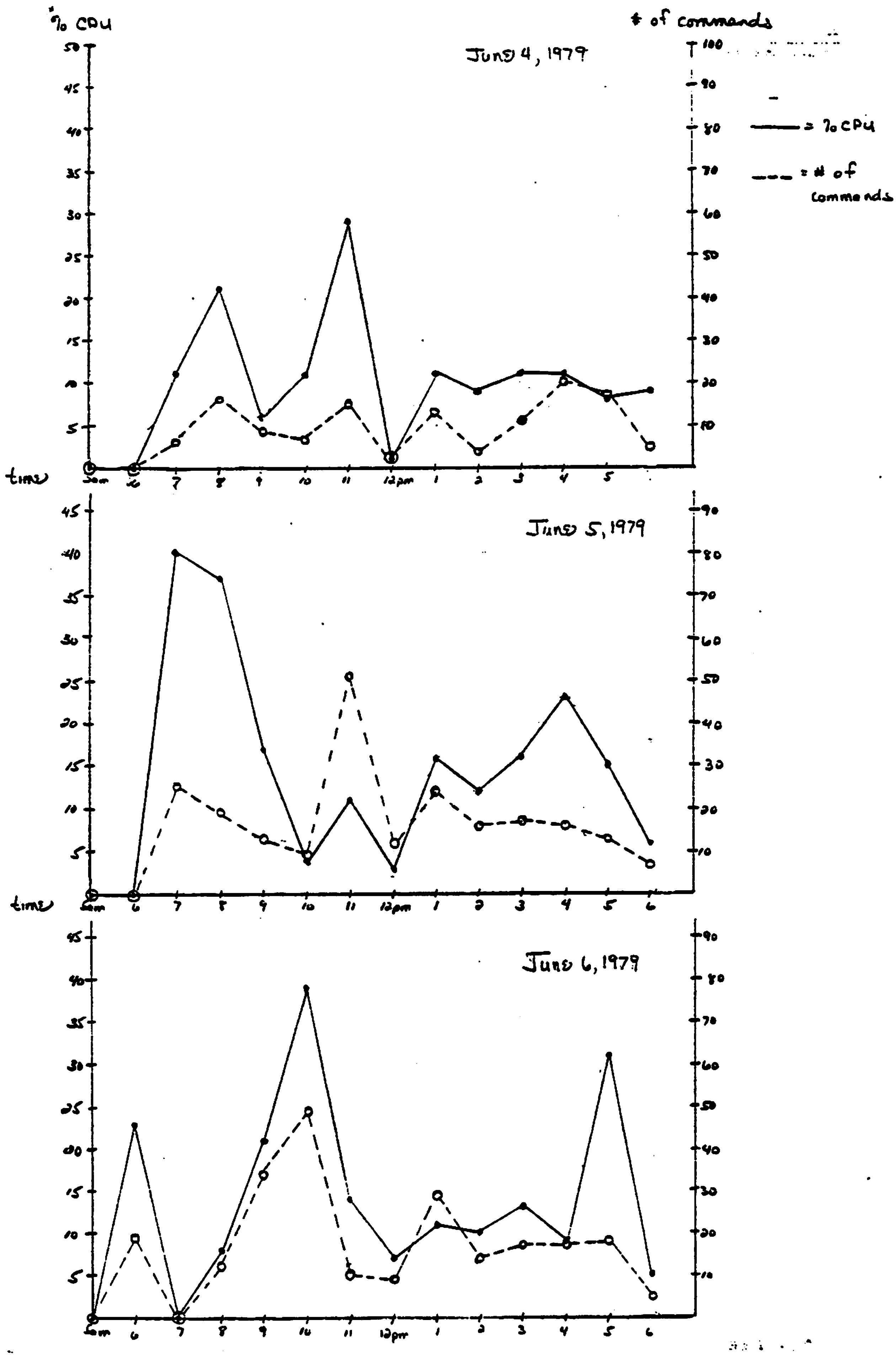
IHPS2 - NETWORKING

Appendix 14.4



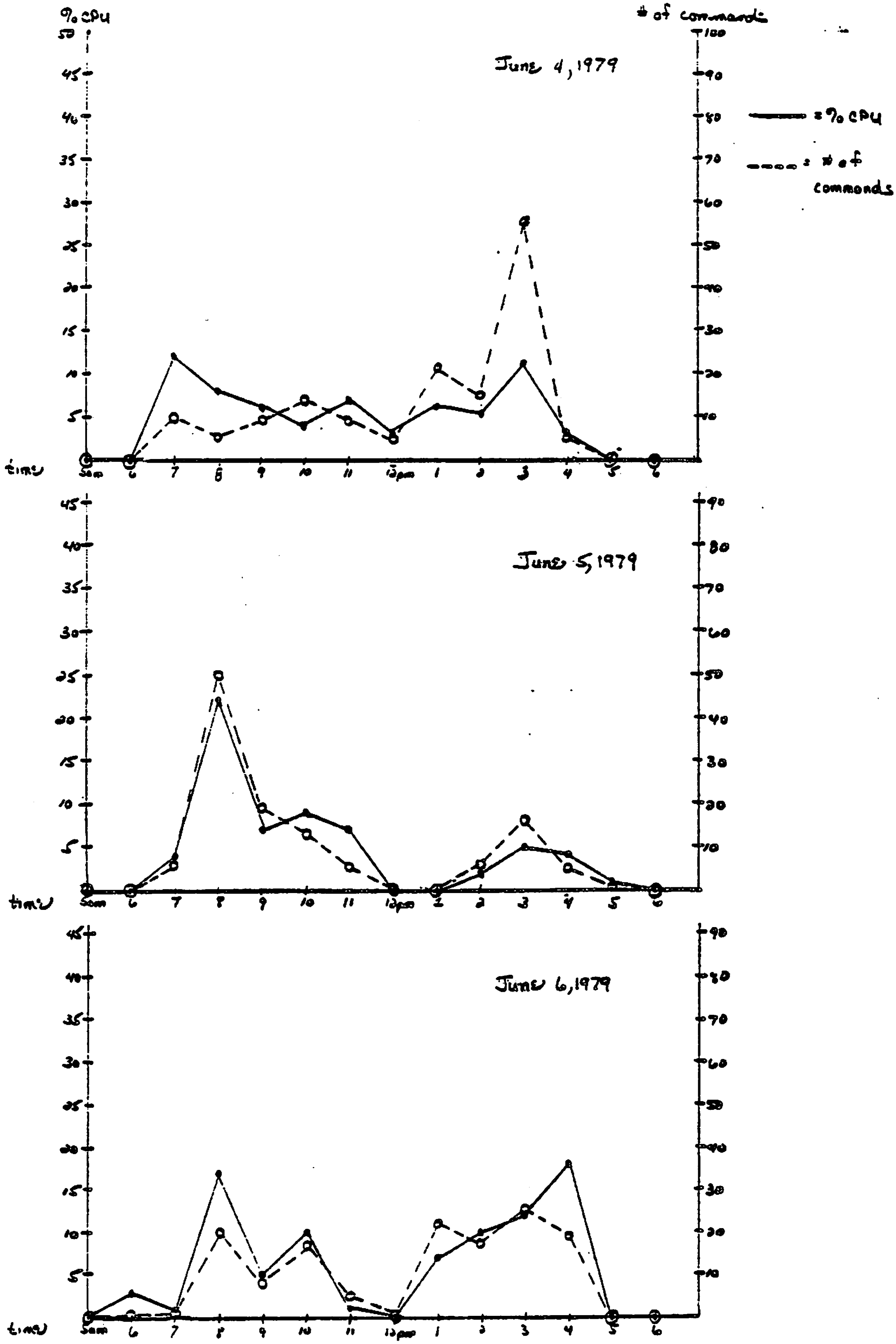
Appendix 15.1

UNIX C - Documentation



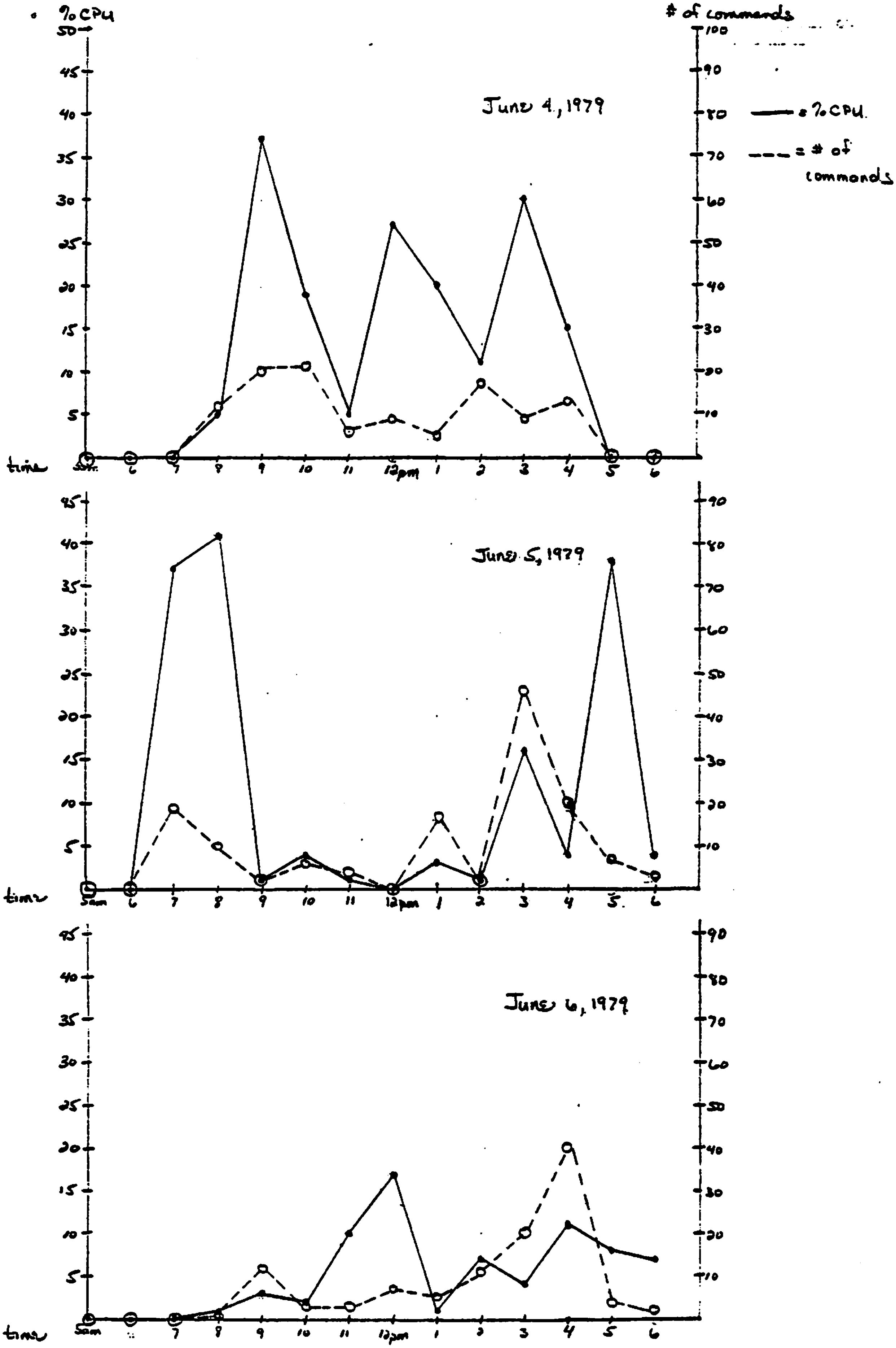
Appendix 15.2

UNIX D - DOCUMENTATION



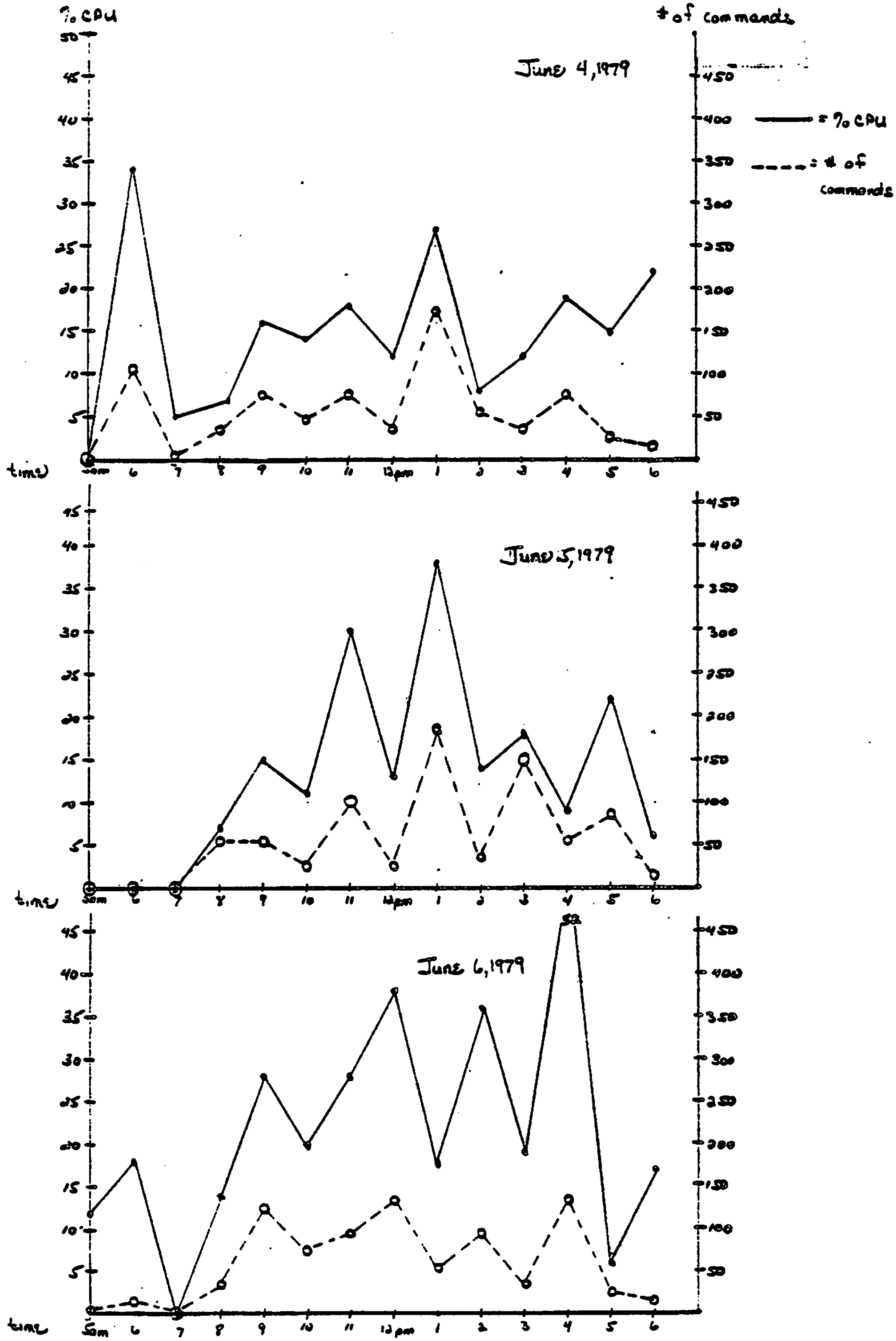
Appendix 15.3

IHFSI - DOCUMENTATION



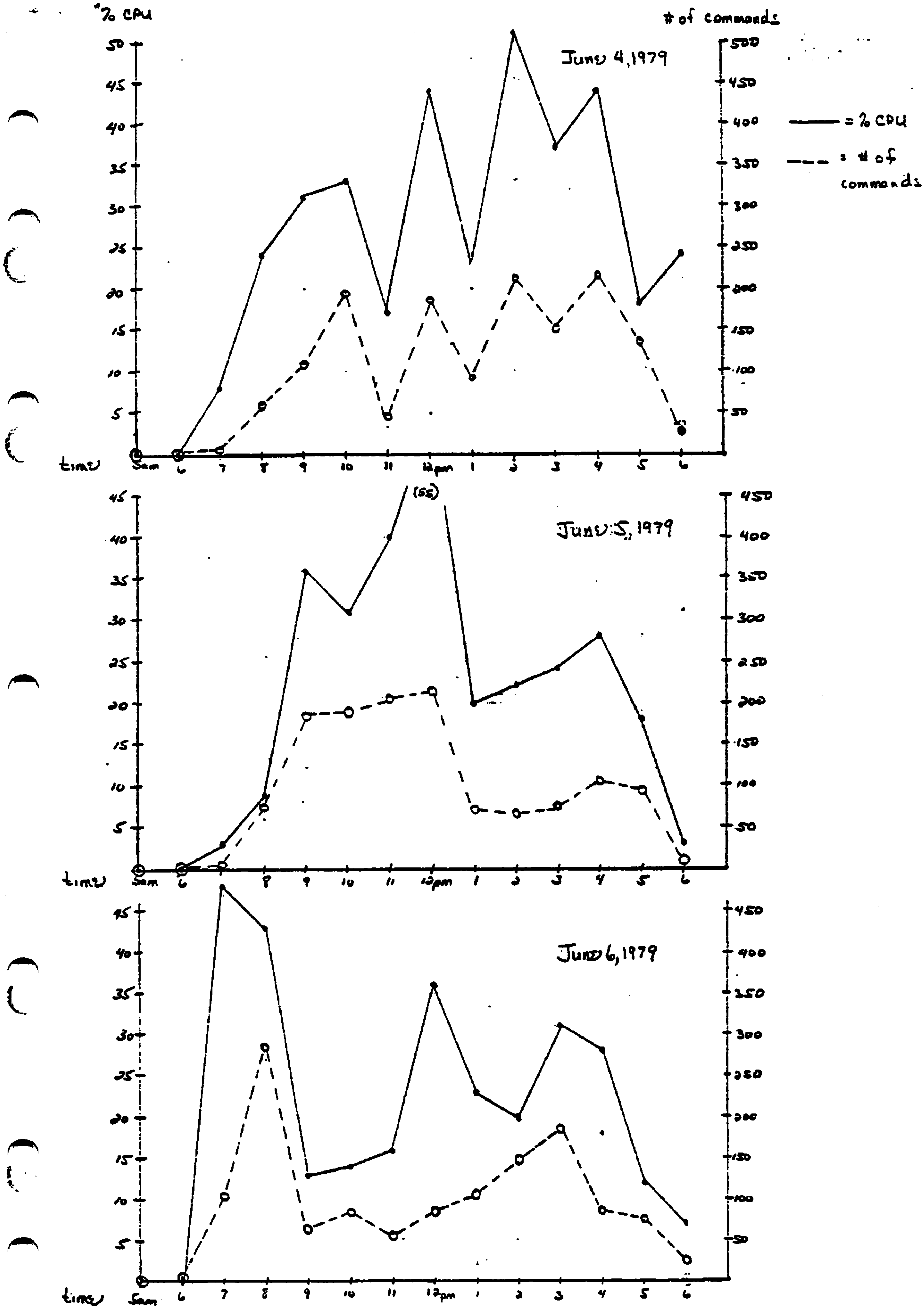
Appendix 15.4

IMPS2 - Documentation



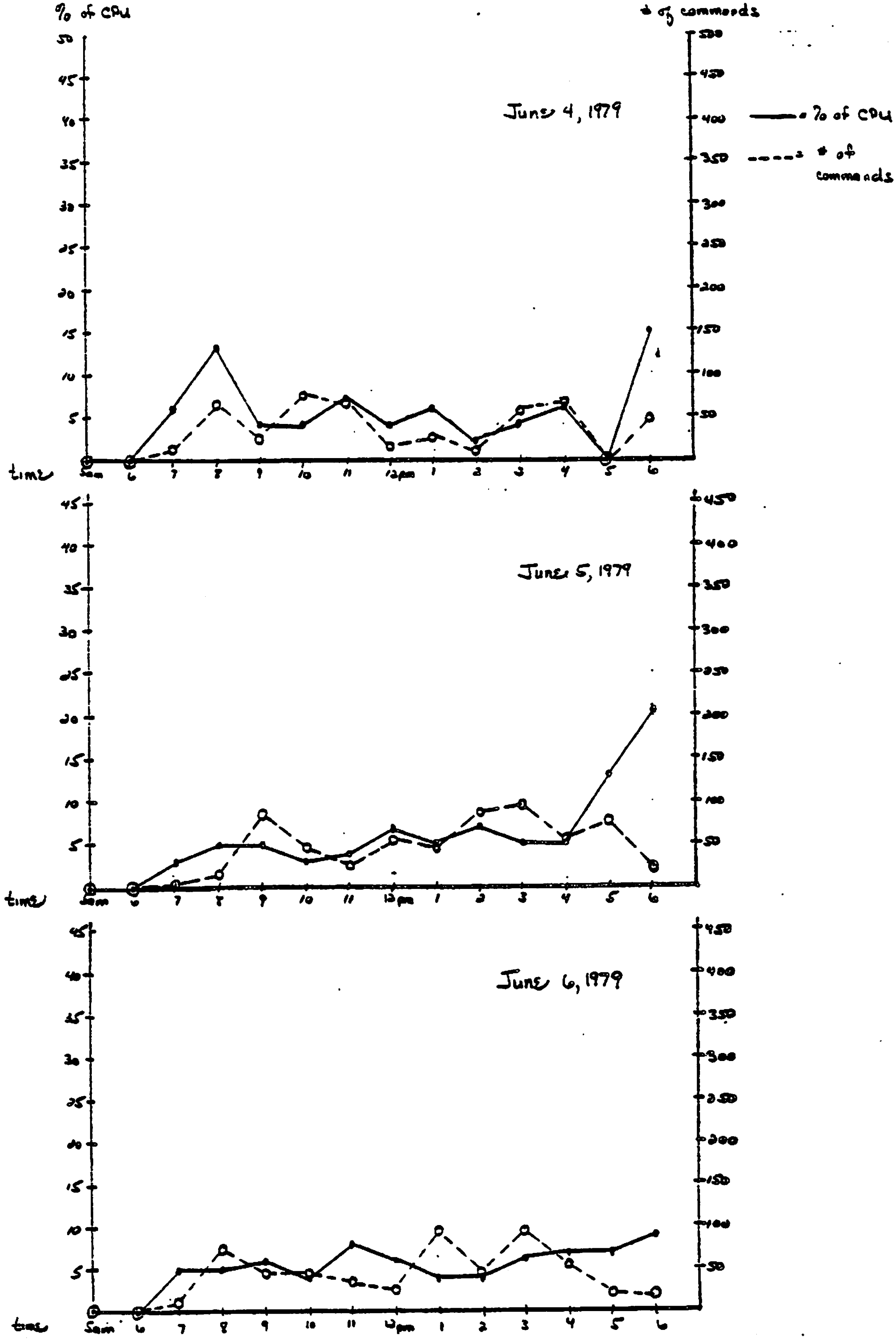
UNIX C - SGS

Appendix 16.1



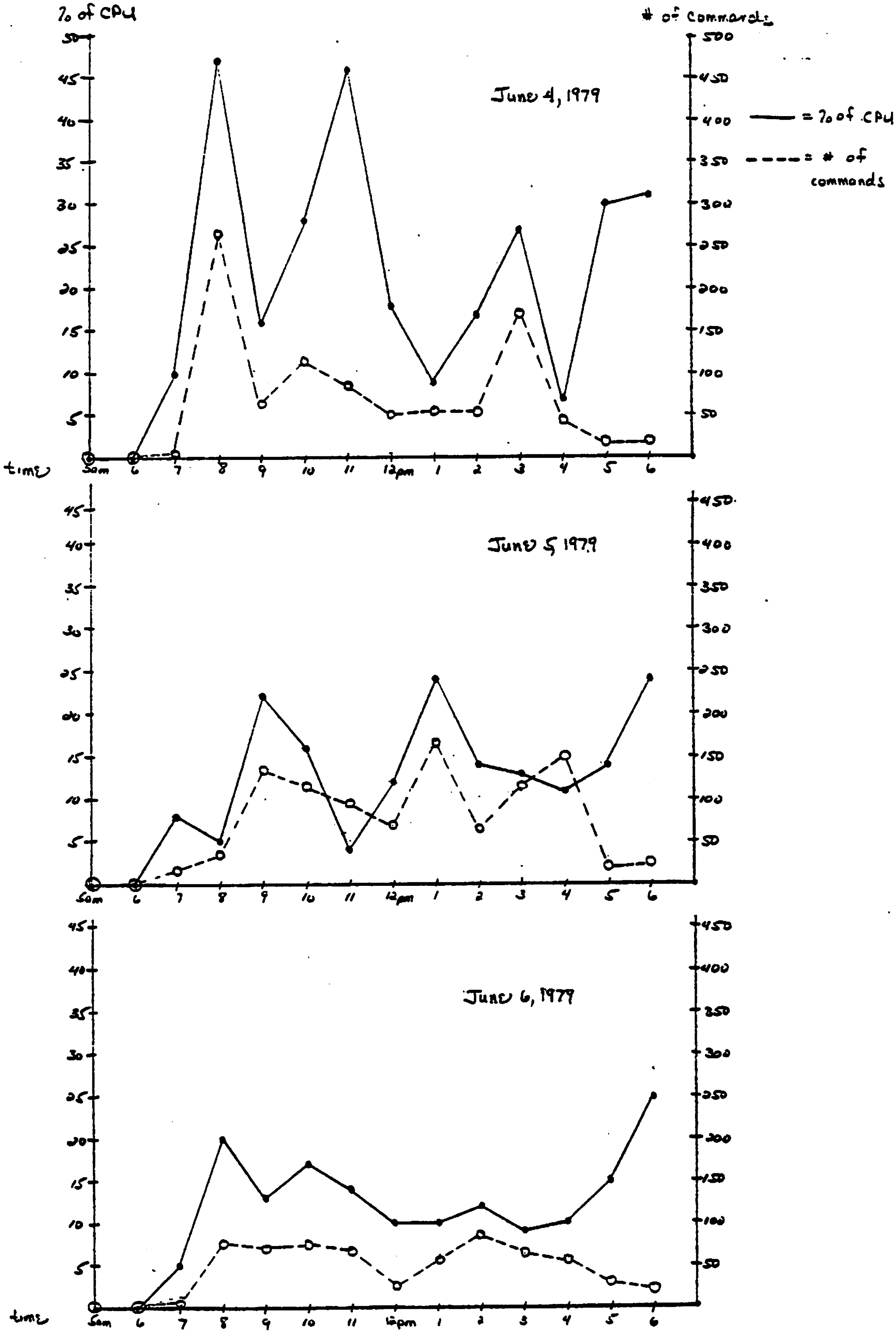
Appendix 16.2

UNIX D - SGS



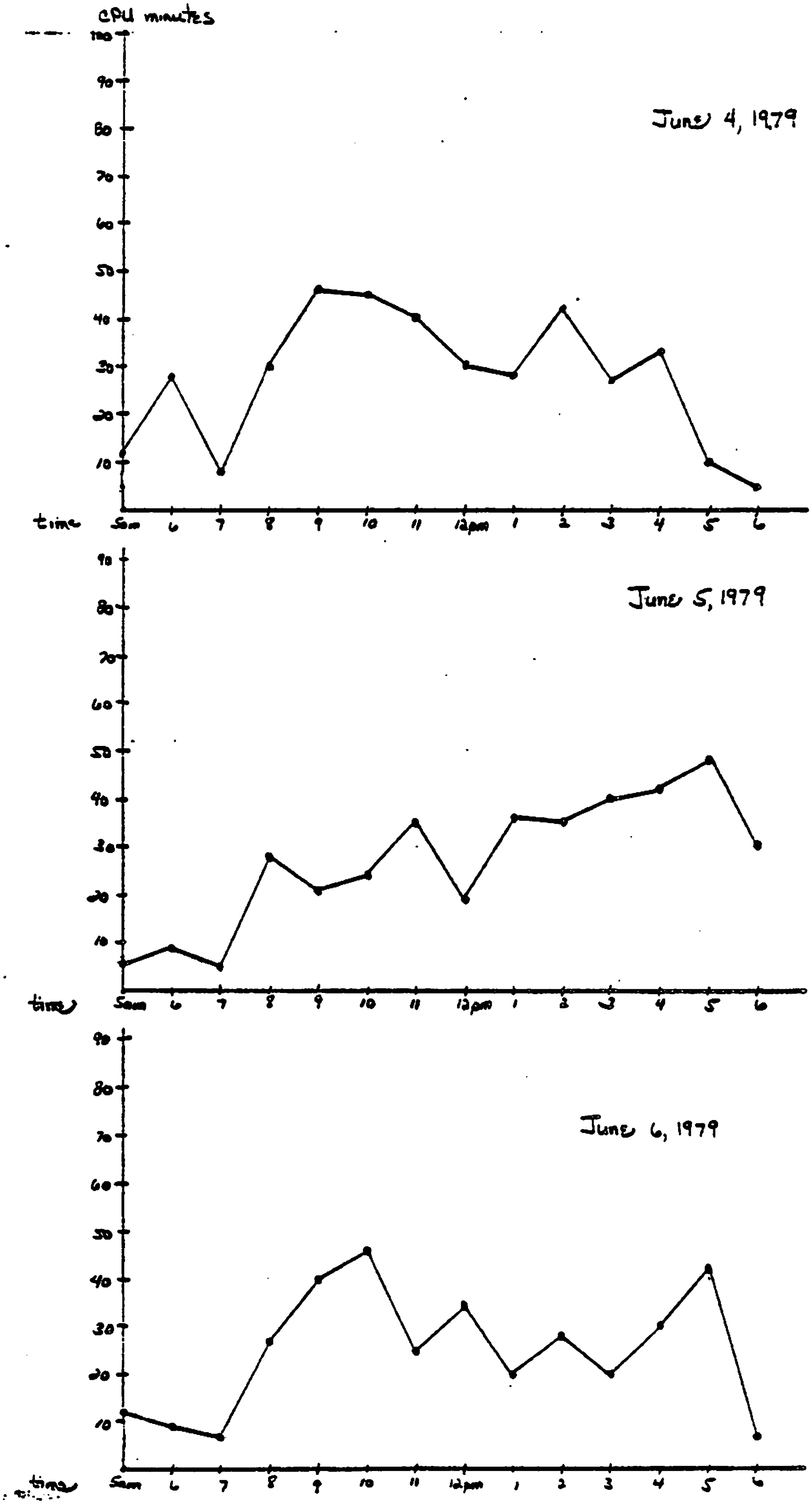
Appendix 16.3

IhPS1 - SGS



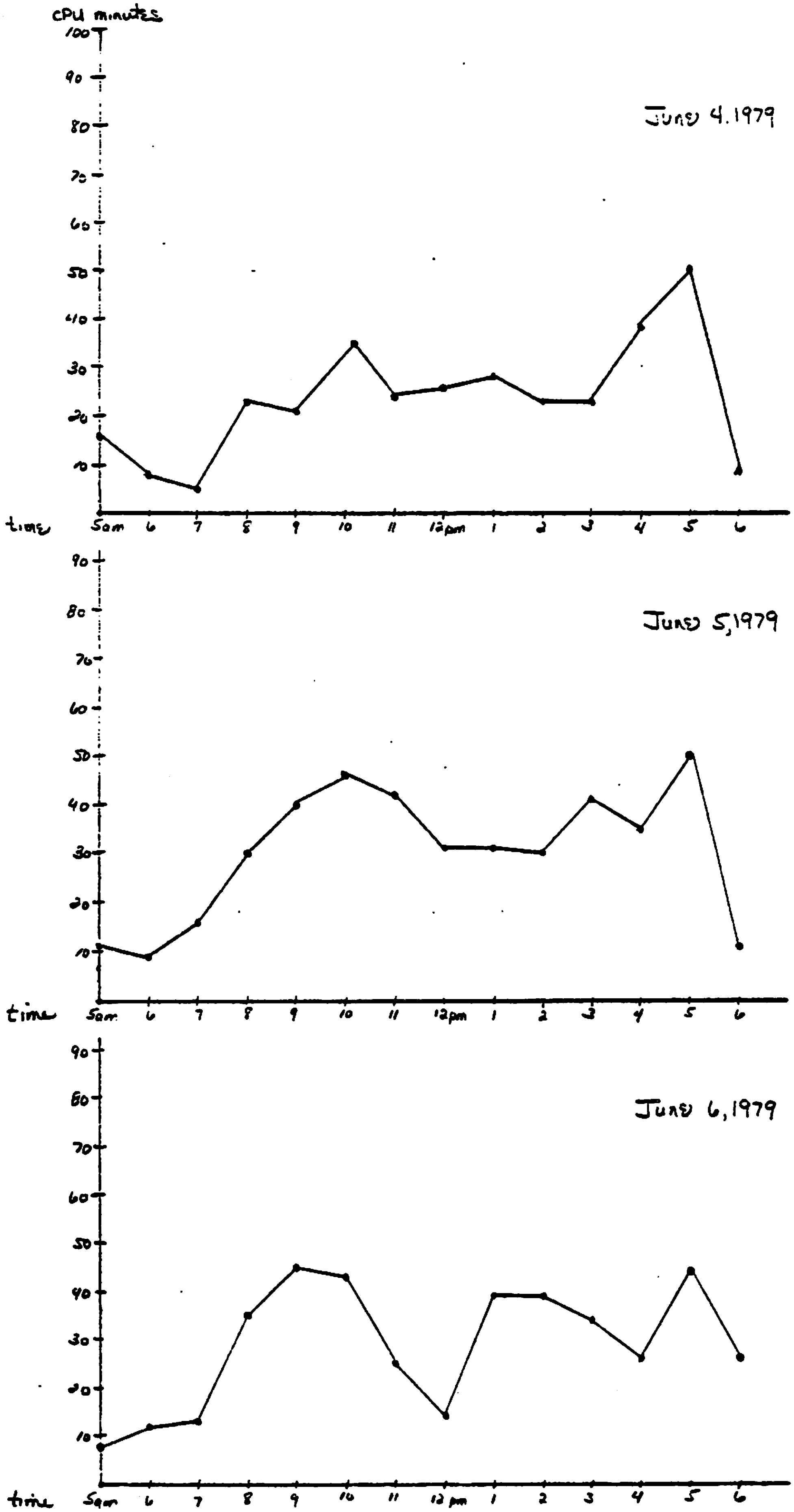
IHPS2 - SGS

Appendix 16.4

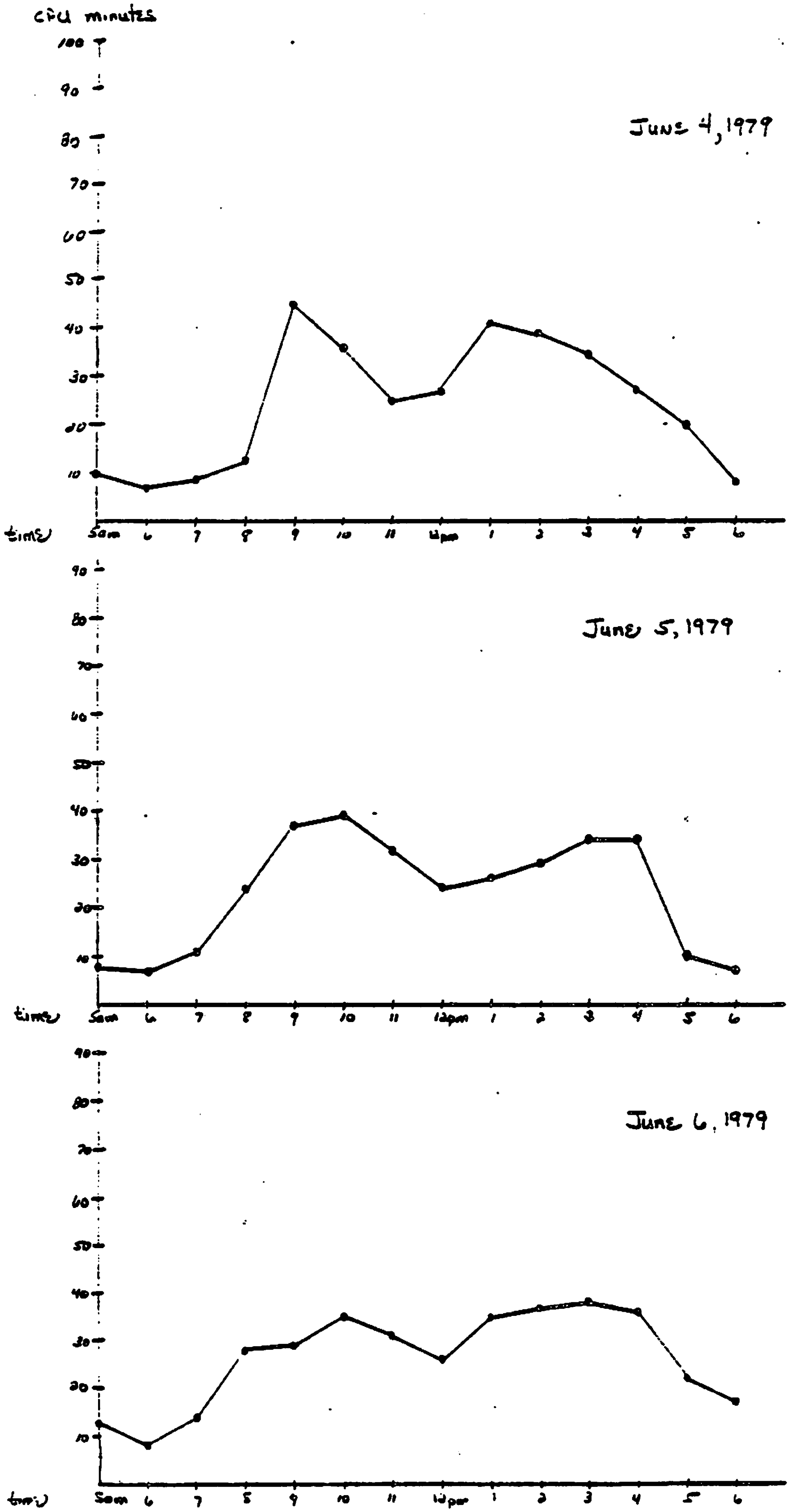


Appendix 17.1

UNIX C - CPU minutes

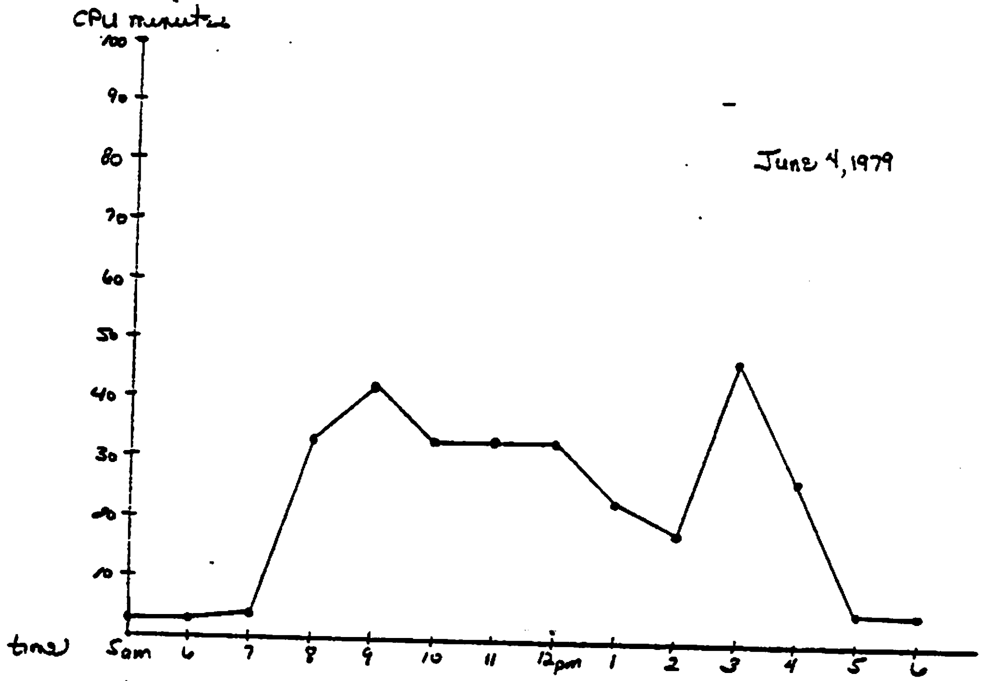


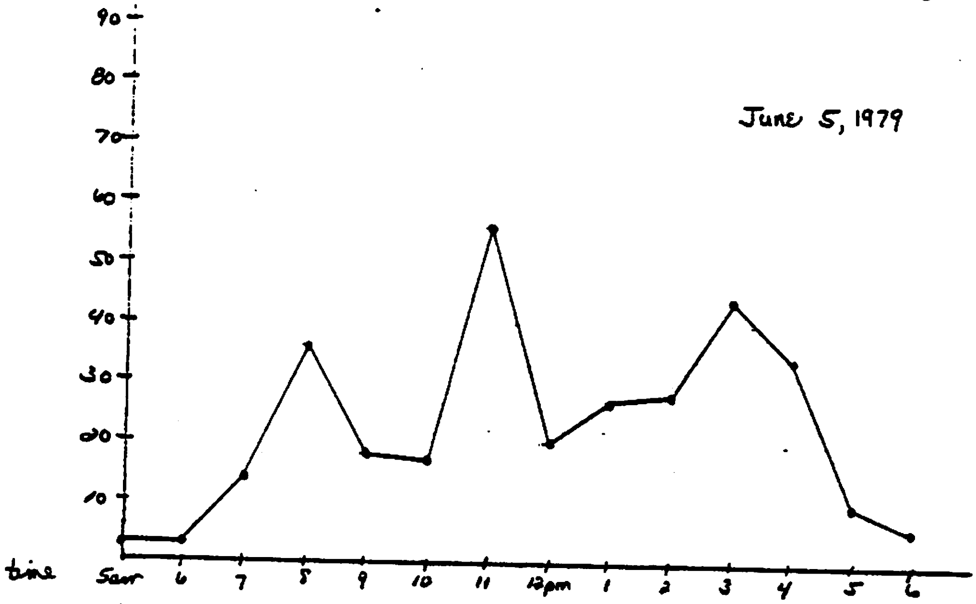
Appendix 17.2 UNIX D - CPU minutes

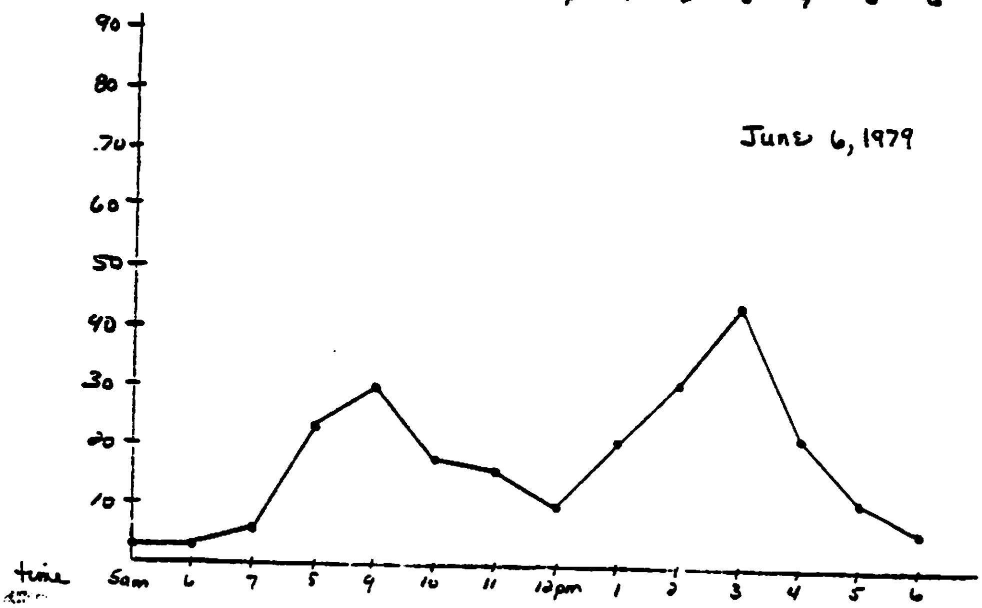


Appendix 17:3

IHPS1 - CPU minutes



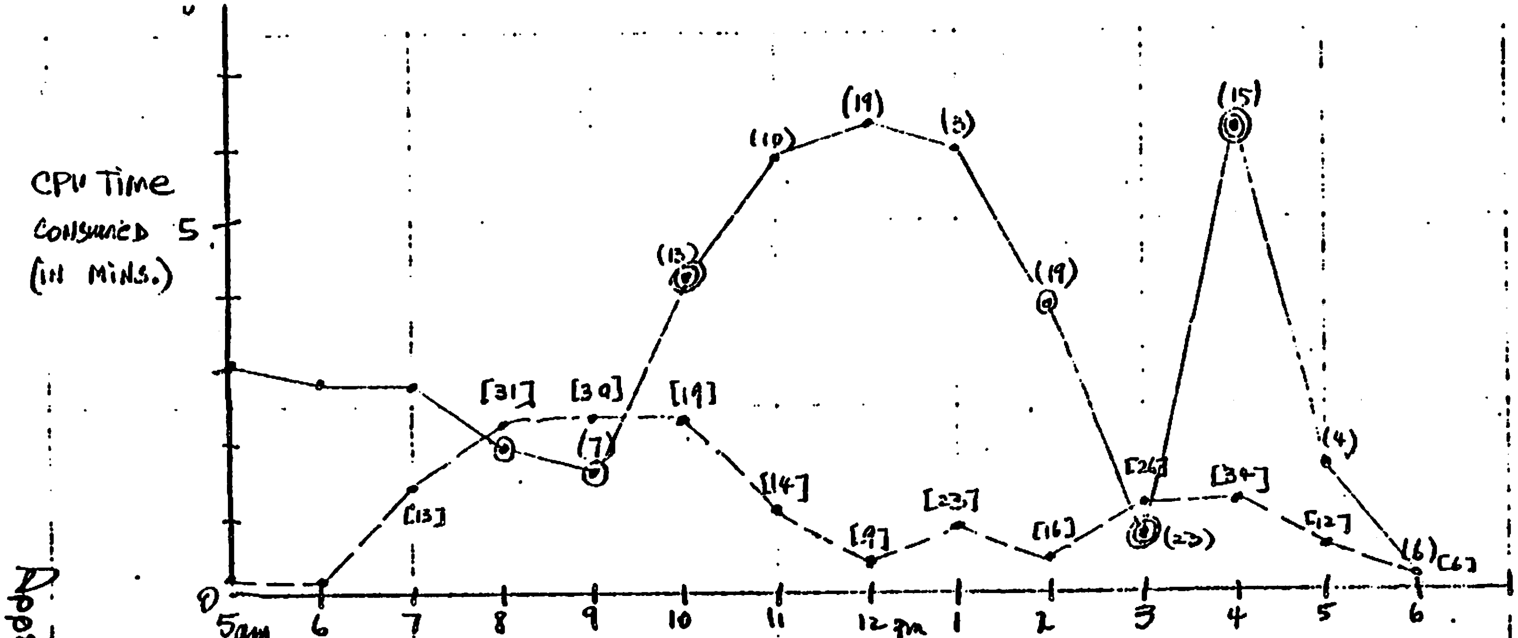




Appendix 17.4

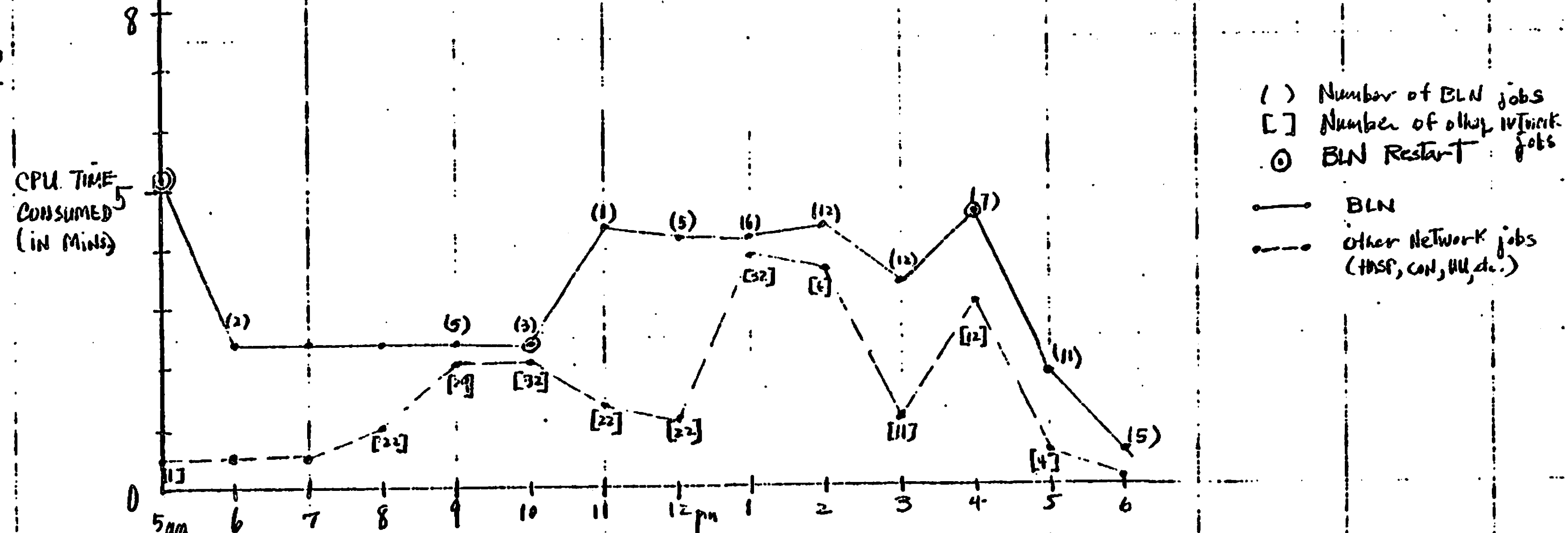
IHPS2 - CPU minutes

Appendix 18.1



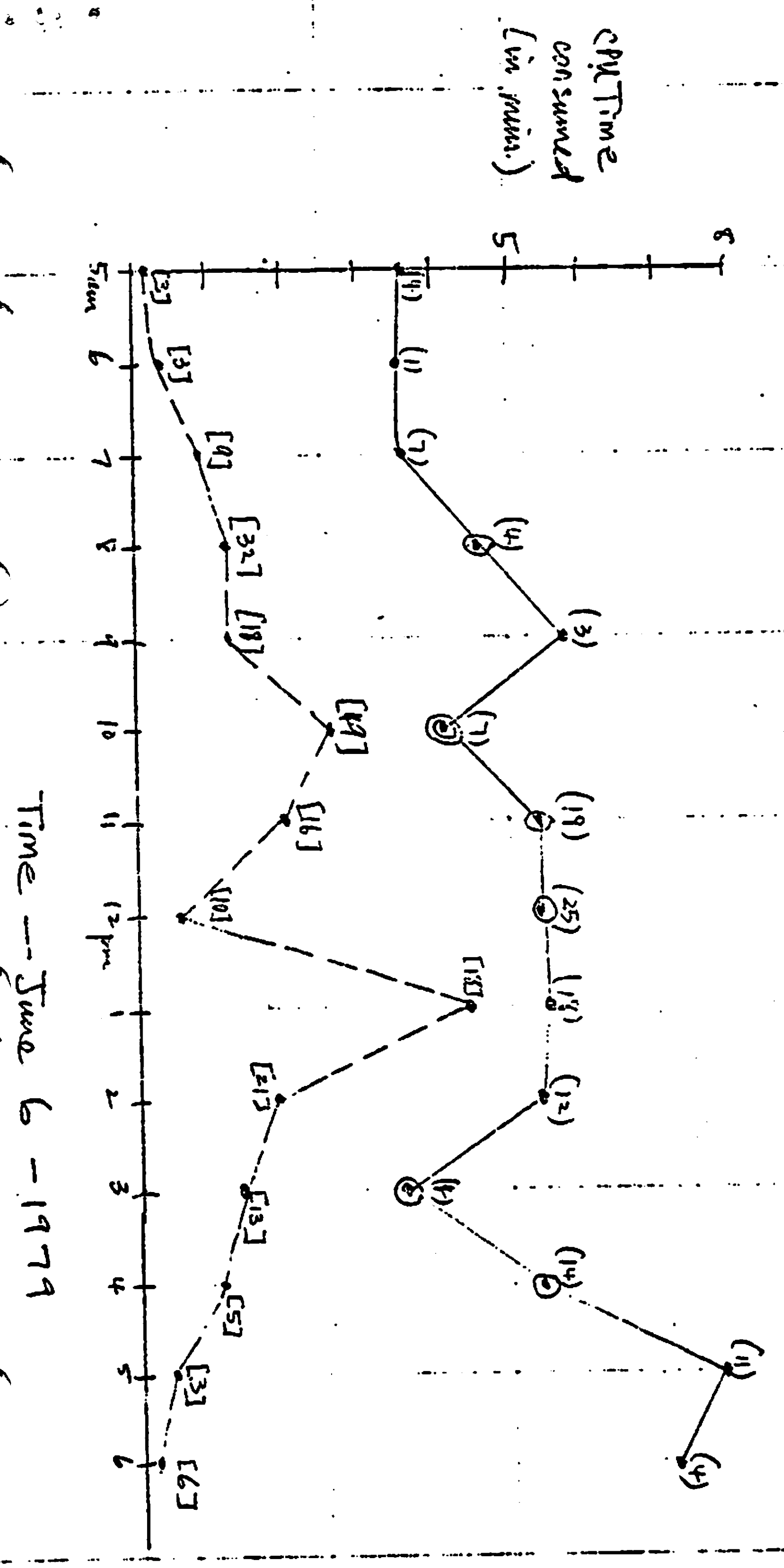
Time – June 5, 1979

PSS 1



Time – JUNE 4, 1979

PSS 1



Appendix 18.2

GREP and SORT on IHPSI

time	0604				0605			
	GREP		SORT		GREP		SORT	
	TCPU (min)	#cmd	TCPU (min)	#cmd	TCPU (min)	#cmd	TCPU (min)	#cmd
5am	.07	26			.07	28		
6am	.08	30	.04	1	.09	30	.08	1
7am	.13	38			.08	27	.	
8am	.09	28			.06	24	.44	26
9am	.1	23	9.1	2	.05	22	8.19	12
10am	.37	61	.06	8	.8	114	.01	1
11am	.07	27	.08	7	3.35	341	.05	3
12pm	.06	22	268	1	1.46	143	.3	3
1pm	.06	25	10.72	4	3.31	335	.1	1
2pm	.08	29	8.15	5	4.12	414	.24	4
3pm	.21	58	3.52	18	3.12	323	.07	6
4pm	.13	40			3.08	298	.08	1
5pm	.09	29			.07	28	.11	2
6pm	.08	28			.06	24		

Commands per Hour - UNIX C

time	0604			0605			0606		
	init	exec	%	init	exec	%	init	exec	%
5am	630	630	100	316	316	100	471	471	100
6am	1015	1062	96	490	535	92	517	561	92
7am	551	602	92	641	685	94	499	542	92
8am	1456	1503	97	1557	1604	97	1629	1672	97
9am	2819	2882	98	1324	1382	96	1601	1658	97
10am	2990	3056	98	1856	1917	97	1160	1221	95
11am	2660	2739	97	1058	1118	95	1771	1822	97
12pm	2177	2237	97	773	830	93	1035	1088	95
1pm	1651	1710	97	1800	1860	97	911	958	95
2pm	1535	1593	96	1716	1776	97	1225	1282	96
3pm	1486	1535	97	2099	2162	97	1402	1458	96
4pm	1469	1521	97	2139	2189	98	1270	1324	96
5pm	701	716	98	839	870	96	693	719	96
6pm	497	501	99	758	779	97	541	551	98

Average percentage of commands totally executed within hour is 96%.

Appendix 20.1

Commands per Hour - UNIX. D

time	0604			0605			0606		
	init	exec	%	init	exec	%	init	exec	%
5am	617	617	100	314	314	100	429	429	100
6am	514	552	93	530	571	93	554	592	94
7am	519	555	94	1131	1171	97	796	833	96
8am	1069	1110	96	1536	1592	96	1487	1534	97
9am	1460	1504	97	1653	1713	96	1516	1572	96
10am	1434	1476	97	1448	1529	95	1353	1411	96
11am	999	1051	95	1233	1306	94	1076	1137	95
12pm	1086	1122	97	1038	1108	94	862	904	95
1pm	977	1020	96	1397	1445	97	1561	1609	97
2pm	984	1024	96	1311	1369	96	1773	1824	97
3pm	1189	1226	97	1587	1636	97	1335	1384	96
4pm	1294	1340	97	1088	1160	94	1259	1311	96
5pm	885	923	96	1000	1051	95	883	918	96
6pm	554	570	97	613	635	97	655	685	96

Average percentage of commands totally executed within hour is 96%.

Appendix 20.2

Commands per Hour - IHPS1

time	0604 init	0604 exec	0604 %	0605 init	0605 exec	0605 %	0606 init	0606 exec	0606 %
5am	684	721	95	707	748	95	813	848	96
6am	672	709	95	577	614	94	616	651	95
7am	725	762	95	851	889	96	1113	1150	97
8am	1075	1114	96	1511	1556	97	1929	1971	98
9am	1382	1424	97	1807	1860	97	1942	1988	98
10am	1824	1870	98	2189	2248	97	2439	2491	98
11am	1488	1534	97	2313	2365	98	2442	2495	98
12pm	1139	1179	97	2008	2047	98	2025	2070	98
1pm	1485	1532	97	1989	2033	98	2559	2615	98
2pm	1593	1653	96	2547	2592	98	2385	2433	98
3pm	2160	2208	98	2585	2637	98	2324	2393	97
4pm	1757	1803	97	2381	2426	98	2229	2288	97
5pm	1178	1204	98	955	981	97	1331	1354	98
6pm	941	951	99	687	693	99	1001	1017	98

Average percentage of commands totally executed within hour is 97%.

Appendix 20.3

Commands per Hour - IHPS2

time	0604			0605			0606		
	init	exec	%	init	exec	%	init	exec	%
5am	349	382	91	325	356	91	326	358	91
6am	367	400	92	332	363	91	324	356	91
7am	421	454	93	764	795	96	493	525	94
8am	1480	1518	97	1371	1407	97	1353	1395	97
9am	1195	1241	96	1217	1258	97	1754	1803	97
10am	1589	1643	97	1120	1165	96	1227	1281	96
11am	1117	1169	96	1127	1169	96	999	1050	95
12pm	955	998	96	1197	1234	97	750	789	95
1pm	988	1031	96	1630	1669	98	1333	1370	97
2pm	1113	1151	97	1763	1805	98	1571	1618	97
3pm	1432	1470	97	1858	1912	97	1960	2028	97
4pm	1427	1470	97	1449	1489	97	1763	1808	98
5pm	461	481	96	515	534	96	899	921	98
6pm	459	469	98	562	591	95	434	448	97

Average percentage of commands totally executed within hour is 96%

Appendix 20.4