1057

Bell Laboratories

# Cover Sheet for Technical Memorandum



Title - Benchmark of UNIX on DEC 11/70

Date - April 17, 1975

TM - 75-8234-4
75-1271-4

Other Keywords - UNIX Operating System
Performance Evaluation



| Author(s) | Location and Room | Extension |
|---|---|---|
| R. B. Brandt | MH 2D-428 | MH 3669 |
| K. Thompson | MH 2C-523 | MH 2394 |



Charging Case - 39199

Filing Case - 40952-001
39199-11


## ABSTRACT

Digital Equipment Corporation (DEC) has announced the 11/70 processor as the newest and most powerful member of the PDP-11 family. A benchmark was run on a prototype 11/70 with some of the newer peripheral I/O devices and compared to a similar benchmark run on an 11/45. Most compute bound programs ran between 2.0 and 2.3 times faster on the 11/70. This improvement was somewhat negated when extensive floating point arithmetic was done. Command response time was found to be 2.5 to 3.0 times better than on the 11/45. Additionally, an unmodified off-the-shelf UNIX system was booted and run on the 11/70.


| | | | | | | Address Label |
|---|---|---|---|---|---|---|
| Pages Text | 12 | Other | 10 | Total | 24 | |
| No. Figures | 1 | No. Tables | 0 | No. Refs. | 1 | |

BELL TELEPHONE LABORATORIES, INC.

TM-75-8234-4

DISTRIBUTION
(REFER GEI 13.9-3)

COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES

OFFICIAL FILE COPY
PLUS ONE COPY FOR
EACH ADDITIONAL FILING
CASE REFERENCED

DATE FILE COPY
(FORM E-1328)

10 REFERENCE COPIES

AHO,A V
ARTHURS,EDWARD
+BALDWIN,G L
BARTLETT,WADE S
BLINN,JAMES C
BRANDT,RICHARD B
BROWN,W STANLEY
CHEN,STEPHEN
CHERRY,MS L L
>CLOUTIER,J E
COOPER,DENNIS W
COPP,DAVID H
CRUM,TED A
DOWD,PATRICK G
FEDER,J
FLEISCHER,HERBERT I
FRASER,A G
FREEMAN,K GLENN
GOLDSTEIN,A JAY
HAMILTON,PATRICIA
HAMMING,R W
+HANNAY,N B
HANSEN,MRS G J
JOHNSON,STEPHEN C
KEESE,W M
KELLY,L J
KERNIGHAN,BRIAN W
KLEIN,MISS R L
LUDERER,GOTTFRIED W R
MARANZANO,JOSEPH F
MC GILL,ROBERT
MC ILROY,M DOUGLAS
+MCDONALD,H S
MORGAN,S P
OSSANNA,J F JR
PEREZ,MRS CATHERINE D
PILLA,MICHAEL A
PINSON,ELLIOT N
+PRIM,ROBERT C
RALEIGH,THOMAS M
RITACCO,J E
+RITCHIE,DENNIS M
+ROCHKIND,M M
SCHLEGEL,C T
>SMITH,D W
TAGUE,BERKLEY A
TERRY,M E
THOMSON,M-L
VAN HAUSEN,J DAVID
VOGEL,G C
WALFORD,ROBERT B

COMPLETE MEMORANDUM TO

YAMIN,MRS E E
52 NAMES

COVER SHEET ONLY TO

CORRESPONDENCE FILES

4 COPIES PLUS ONE
COPY FOR EACH FILING
CASE

AMOSS,JOHN J
ARCHER,RUSSELL E JR
ARNOLD,GEORGE W
ARNOLD,S L
BAKER,BRENDA A
BAYER,DOUGLAS L
BECKER,R A
BERTH,R P
BIREN,MRS IRMA B
BLUE,J L
BLUM,MRS MARION
BOURNE,STEPHEN R
BOYCE,W M
BROWN,COLIN W
BUCHSBAUM,S J
CALESSO,GIULIO L
CANADAY,RUDD H
CARDOZA,WAYNE M
CARRAN,J H
CARR,DAVID C
CHAMBERS,J M
CHRIST,C W JR
COLE,LOUIS M
COOK,THOMAS J
COSTON,WALTER P
COULTER,J REGINALD
CUNNINGHAM,STEPHEN J
CUTLER,C CHAPIN
DAVIS,D R
DE JAGER,D S
DOLOTTA,T A
EDELSON,D
EDMUNDS,T W
EITELBACH,DAVID L
ELLIOTT,R J
ERRICHIELLO,PHILIP M
ESTOCK,R G
FELDMAN,STUART I
FELS,ALLEN M
FLYNN,MS M L
FORTNEY,MRS VIRGINIA J
FRANK,H G
FREEDMAN,M I
FREEMAN,R DON
FREIDENREICH,MRS B
GATES,G W
GERGOWITZ,MRS ELISE B
GILLETTE,DEAN
GLASER,W A
GLASSER,ALAN L

COVER SHEET ONLY TO

GNANADESIKAN,R
GORMAN,JAMES E
GRAHAM,R L
GRAVEMAN,R F
HAIGHT,R C
HALL,ANDREW D JR
HARKNESS,C J
HAWKINS,DONALD T
HELD,RICHARD W
HOYT,WILLIAM F
HUMCKE,D J
HUPKA,MRS FLORENCE
HYMAN,B
IMAGNA,CLYDE P
IVIE,EVAN L
JENSEN,PAUL D
JUDICE,CHARLES N
KAISER,J F
KAPLAN,A E
KAUFMAN,LARRY S
KETCHLEDGE,B A
KNOWLTON,KENNETH
KNUDSEN,DONALD B
LA CAVA,JOSEPH L
LAMBERT,MRS C A
LANDAU,C N
LESK,MICHAEL E
LESSEK,PETER V
LEVINE,P R
LIAUGAUDAS,
LICWINKO,J S
LIST,WILLIAM H S
LUMMIS,ROBERT C
LYCKLAMA,HEINZ
LYONS,T G
MACHOL,R E JR
MASHEY,JOHN R
MASHEY,MRS R
MATHEWS,MAX V
MC CULLOUGH,RICHARD H
MC MILLAN,W F
MC TIGUE,G E
MENNINGER,R E
MENZEL,MRS F S
MILLER,S E
MILLSPAUGH,J A JR
MILNE,D C
MOLINELLI,JOHN J
MORGAN,DENNIS J
MORRIS,ROBERT
MORR,PHILLIP L
MUSA,J D
NETZEL,T L
NIMTZ,R O
NINKE,WILLIAM H
NOLL,J C
NORTON,MRS SUZANNE W
NOWITZ,D A
O CONNELL,T F
ORCHARD,R A
PALUCK,MRS M B
PATEL,C K N
PEKARICH,S P

COVER SHEET ONLY TO

PERDUE,R J
PERNESKI,A J
PETERSON,RALPH W
PETERSON,T G
PHILLIPS,S J
PLAUGER,P J
POLICH,MISS KAREN K
POLLAK,HENRY O
POLLOCK,DAVID A
POPPER,C
RAFSKY,L C
ROBERTS,CHARLES S
ROCHKIND,M J
RODIHAN,MRS PATRICIA A
ROM,DOUGLAS B
ROOME,WILLIAM D
ROSENTHAL,CHARLES W
ROSLER,LAWRENCE
RYDER,BARBARA G
SATZ,L R
SCHRYER,N L
SCHWENKER,JOHN E
SCRIBNER,NEAL M
SIDOR,DONALD J
SILVERSTEIN,STEVEN M
SIX,FREDERICK B
SOUTHERN,MS M J
SPENCE,NORMAN A
STROHECKER,CARY A
STUCK,BARTON W
STURMAN,JOEL N
SWANSON,GEORGE K
TANIS,MRS PATRICIA J
THOMPSON,BERNARD E
THOMPSON,K
THOMPSON,MRS M
TILLOTSON,L C
TUKEY,JOHN W
VADIMSKY,R G
VAN ORNUM,JAMES H
VARNEY,ROBERT C
VIGGIANO,F A
VOLLARO,JOHN R
WALKER,MISS E A
WANDZILAK,P D
WARNER,D D
WATKINS,G T
WEBB,FRANCIS J
WEHR,L A
WHIPPLE,JOHN H
WHITE,RALPH C JR
WILHOYTE,W L
WILSON,M P
WINCKELMANN,W A
WINHEIM,MISS IRENE A
WOOD,J L
ZEPP,HILTON E
170 NAMES

+ NAMED BY AUTHOR

> CITED AS REFERENCE SOURCE

MERCURY SPECIFICATION.....................................................................................

COMPLETE MEMO TO:
127-SUP 8234-AMTS

COVER SHEET TO:
12-DIR 13-DIR 127
UNHA = UNIX/HARDWARE

---

HARDISON,D
MH 2C121

TM-75-8234-4
TOTAL PAGES 24

TO GET A COMPLETE COPY:

1. BE SURE YOUR CORRECT ADDRESS IS GIVEN ON THE OTHER SIDE.
2. FOLD THIS SHEET IN HALF WITH THIS SIDE OUT AND STAPLE.
3. CIRCLE THE ADDRESS AT RIGHT. USE NO ENVELOPE.

PLEASE SEND A COMPLETE COPY TO THE ADDRESS SHOWN ON THE OTHER SIDE
NO ENVELOPE WILL BE NEEDED IF YOU SIMPLY STAPLE THIS COVER SHEET TO THE COMPLETE COPY.
IF COPIES ARE NO LONGER AVAILABLE PLEASE FORWARD THIS REQUEST TO THE CORRESPONDENCE FILES.

Bell Laboratories

subject: Benchmark of UNIX on DEC 11/70

date: April 17, 1975

from: R. B. Brandt
K. Thompson
TM-75-8234-4
TM-75-1271-4

MEMORANDUM FOR FILE

## Introduction

Digital Equipment Corporation (DEC) has announced the 11/70 processor as the newest and most powerful member of the PDP-11 family. In a nutshell, the 11/70 is an 11/45 processor with cache memory, a capability for up to 2M words of memory, and a new bussing structure for certain high speed I/O devices (the so-called Mass Bus). On January 24-26, 1975, R. B. Brandt, J. F. Maranzano, and T. M. Raleigh of the UNIX Support Group and K. Thompson of the Computing Techniques Research Department ran a benchmark on a prototype 11/70 at DEC's Maynard, Massachusetts, plant. This memorandum describes the results of that benchmark and a similar benchmark that was run on the UNIX Support Group's 11/45.

## Conclusions

The results of these tests indicate an 11/70 performance increase over an 11/45. However, the exact amount of increase will vary according to a system's configuration and load. The question of what is the load on a system is usually imprecise at best, varying at most development oriented installations from day to day and even from hour to hour. However, the following specific conclusions about the 11/70's performance were reached based on the tests run during this benchmark.

1. The speed of the 11/70 processor itself was found to be between 1.3 and 2.7 times faster than the 11/45. Almost all of the compute bound programs that did not extensively use floating point ran between 2.0 and 2.3 times faster on the 11/70. This improvement is due to the addition of the cache.

2. It is worth noting that heavy use of floating point will negate some of the performance improvement attributed to the 11/70 processor. The

compute bound programs that did a great deal of floating point arithmetic ran only 1.3 to 2.0 times faster on the 11/70 as on the 11/45.

3. As far as response time is concerned, the 11/70 configuration tested showed an improvement in command script elapsed times that ranged from 1.6 to over 4 times that of the 11/45 tested. This is indeed a wide range, but the endpoints here may be misleading. The majority of the tests were in the 2.5 to 3.0 range, and it is fair to say that this is a valid range to use when discussing response differences. This performance improvement can be attributed to three major differences between the 11/70 and the 11/45 tested: the processor itself, the newer peripheral devices on the 11/70, and the additional memory available on the 11/70.

4. The addition of main memory should result in improved response on either processor. It is not clear at this time how much improvement would result from adding memory to the 11/45 configuration. This is a question that can be answered only through further testing.

5. A slight improvement in response was attainable by having memory simulate a peripheral I/O device. However, this would not be cost effective on any production or development system. If the money is to be spent on additional main memory for a system, it would be better to not use that memory for simulation of peripherals.

A problem that was left unresolved occurred intermittently during 11/70 testing. The floating point hardware would cause halts in user mode, which is normally an impossible situation. The DEC engineers on-site were unable to explain why this happened.

An unmodified UNIX will run on an 11/70. However, it would be advantageous to make modifications to the system in order to handle two particular situations. First, the standard UNIX does not currently use the 11/70's Memory management Unit in 22 bit mode, the Unibus Map, or the address extension registers in the high speed controllers. This restricts the physical address space of UNIX to 128K words. If an 11/70 is to be installed with more than 128K words of true memory, these modifications would have to be made in order to utilize the additional memory. And second, the cache memory on the 11/70 has introduced the possibility of cache parity errors. Several of these occurred during 11/70 testing and caused crashes, because UNIX was not programmed to handle the resultant trap. The appropriate modifications should be made to system software so that error recovery is attempted whenever possible. None of the aforementioned UNIX modifications are difficult, but the changes necessary to accommodate the Unibus Map can not be made in a clean manner.

The 11/70

Before proceeding with an analysis of benchmark results, a brief description of the 11/70 is in order. This description is not intended to fully explain this processor, but, rather, will only highlight the major new features. More detailed information may be obtained from either reference [1] or your local DEC salesman. This discussion of the 11/70 presupposes that the reader has at least a rudimentary knowledge of the 11/45's architecture and terminology.

The 11/70's CPU assembly is essentially an 11/45 processor with some additional hardware attached to improve performance. Figure 1 is a block diagram of the 11/70's architecture. The Memory Management Unit (MMU) is an integral part of the 11/70 processor, whereas it was optional on the 11/45 (although a MMU was required if more than 28K words of true memory were used). The 11/70's MMU operates in a manner similar to that of the 11/45 MMU in that, when enabled, 16 bit program addresses are mapped into physical memory addresses. However, the 11/45 MMU only translates program addresses into 18 bit physical addresses, so that the physical address space is limited to at most 128K ($2^{17}$) words. The 11/70 MMU will translate program addresses into either 18 bit or 22 bit real addresses. The 18 bit mode preserves upward compatibility from the 11/45, while the 22 bit mode provides the potential for a 2M word physical address space. (DEC currently limits 11/70 memory size to 1M words.) When the MMU is operating in 22 bit mode, references to the upper 128K words of the address space are directed to the Unibus rather than physical memory. Also, the Unibus Map translates 18 bit Unibus addresses into 22 bit real memory addresses, so that peripherals on the Unibus may correctly reference all of the possible real memory locations.

More important for performance than the capability for a large real memory size is the addition of a cache on 11/70 processors. In particular, the 11/70 cache contains 1K words of high speed bipolar memory organized as two word blocks, and utilizes a two set associative scheme. Whenever a memory read fails to find the required data or instruction word in the cache, the even/odd word pair containing the desired word (32 bits total) are transferred from main memory to the cache across a 32 bit bus. It should also be noted that in 11/70 architecture, main memory references do not take place across the Unibus as they do for the 11/45, but, rather, a separate data bus is provided for this purpose. DEC predicts that typical programs will have a cache read hit rate (defined as the ratio of memory reads found in the cache to total memory reads) in the 80 to 95% range and that 90% of all memory references are reads. These numbers have been verified for some important UNIX programs.

Another major performance improvement for the 11/70 is the capability to utilize up to four high speed controllers for certain DEC mass storage peripherals. In particular, the devices that may be used with these (optional) controllers are the RS04 (RS03) Fixed Head Disk, the RP04 Moving Head Disk, and the TU16 Magnetic Tape Drive. The performance improvement potential from use of these high speed controllers is due to decreased contention for the Unibus and the 32 bit data path between memory and the controller.

## Benchmark Objectives

The basic intent of the benchmark was to gain a familiarity with the 11/70, evaluate its capabilities, and exercise some of the newer DEC peripheral devices. A wide variety of peripherals were made available by DEC on the prototype machine. Attachment A details the hardware available on the 11/70 and 11/45 systems evaluated. Each test was run to satisfy one of the following three specific objectives:

1. To determine if unmodified off-the-shelf UNIX systems would run on an 11/70.
2. To test and, if necessary, debug drivers for some of the newer DEC peripheral devices.
3. To measure the performance of the 11/70 in comparison to that of an 11/45.

The first of these objectives was essentially a test of the upward compatibility of UNIX software written for the 11/45 (and 11/40). It was hoped, of course, that this software would run unmodified on an 11/70 system. In the event that it would not, a careful determination of the required changes was to be made so that support could be given to installations acquiring 11/70 processors.

As far as device drivers are concerned, the RP04 disk driver [2] had already been tested and debugged on an 11/45 system prior to this benchmark. However, there was a need to verify that it would perform properly when used in conjunction with one of the high speed controllers unique to 11/70 systems. A driver written for the RS04 (RS03) disk had never been tested. Since the 11/70 benchmark system had this device, it was a good opportunity to test it. In addition, a driver for the TU16 tape drive was to be written and tested.

To evaluate the 11/70's performance, a potpourri of timing tests had been developed prior to the actual benchmark. These tests were designed to evaluate the speed of the processor itself, the speed of the peripheral devices on the high speed controller, and the effect of increasingly heavy loads on the system. Insofar as possible, these same tests were to be run on an 11/45 and the results compared.

## Benchmark Test Results

There is little that needs to be said with respect to the tests run to satisfy the first two benchmark objectives. Both an RK and an RP off-the-shelf UNIX system were booted and run without any need for modification. Any installation upgrading to an 11/70 should, therefore, be able to run the basic UNIX system unmodified on their new CPU. During all of the performance evaluation testing, both the RP04 and the RS04 (RS03) drivers were extensively exercised. Both performed admirably and without failure. A TU16 driver was written and tested, although not extensively. A memory driver was also written to cause a portion of main memory to simulate an I/O device. This driver was used to utilize memory as both the Root device and as the device for TMP (temporary) files in some of the performance tests.

A wide variety of timing tests were run throughout this benchmark on a number of different system configurations. The principal differences between configurations of the same CPU are in the I/O devices included and, in the case of the 11/45, the number of system I/O buffers available. The exact configurations used are given in Attachment B. Rather than explicitly restate the complete configurations in the following test descriptions, the numerical designation assigned in Attachment B will be used. A UNIX system with separate I and D space was used throughout the benchmark. Additionally, for the purposes of the benchmark, the number of I/O buffers in the 11/70 configuration was increased to forty. The 11/70's Memory Management Unit was run in 22 bit mode, but although 640K words of memory were available, the physical address space used by UNIX was artificially limited to 128K words. This was done to circumvent the problems associated with using more than 128K words of memory (ie. 22 bit addresses) in conjunction with peripheral devices. In particular, this problem arises when using the swap device and during "raw" I/O. When used with Unibus devices, memory addresses larger than 18 bits require the setting of mapping registers for use by the Unibus Map in address relocation. To accomplish this function for devices on high speed controllers, a 6 bit address extension register in the controller must be accessed and set.

In timing the various tests, it was possible to measure real (i.e. wall clock) time accurately only to the nearest second and CPU time accurately only to the nearest 1/60 second. This is due to the granularities of the system clocks. Rather than present the timing information for each run separately, only the average times for analogous runs will be given since the results were consistent over the tests. Except as noted otherwise, all times will be expressed in the format minutes:seconds.

1. CPU Bound Programs

These tests were run to evaluate the speed of the 11/70 processor without (or at least, with a minimum of) I/O interference (which includes, of course, swapping). For this reason, each program or shell procedure was executed single thread. In addition, because the primary difference between configurations of the same processor was in I/O related areas and very minimal I/O was involved during actual timing, these tests were run on only one configuration of each processor.

By far, the longest running of any of the compute bound tests was the chess shell procedure. This procedure was written to play both sides of a sixty-five move game (130 moves total) of chess. The program used within the procedure was /usr/games/chess, which is found on many UNIX systems. This program was used because it does no floating point arithmetic. Timing was accomplished by using the "time" command on the shell procedure. The results of this test are summarized in Table 1. It may be observed from these results that the ratio of 11/45 user CPU time to 11/70 user CPU time for this particular test is 2.2.

| Config | Real | User CPU | Sys CPU |
|---|---|---|---|
| C1 (11/45) | 29:09 | 28:53.1 | 11.5 |
| C4 (11/70) | 13:16 | 13:06.1 | 6.3 |

Table 1

After running the chess program, which uses no floating point, a series of programs were run to made heavy use of the floating point processor. These programs were written adhering to the principles of the Gibson Mix [3]. In every instance but one, two equivalent versions of each program were run, one being written in C and the other in Fortran. Each of these programs contained a loop consisting of the particular mix of operations, including subroutine calls, being timed. Timing was done by having the program invoke the appropriate system calls, and the average CPU time (in milliseconds) required for a single execution of the loop was printed. Table 2 details the results of this test (all times are in milliseconds).

| Program | C1(11/45) | C4(11/70) | Ratio |
|---|---|---|---|
| 1 (Fort) | 3.3 | 2.4 | 1.4 |
| 2 (C) | .18 | .14 | 1.3 |
| 2 (Fort) | .68 | .45 | 1.5 |
| 3 (C) | 640 | 480 | 1.3 |
| 3 (Fort) | 2080 | 1300 | 1.6 |
| 4 (C) | .56 | .42 | 1.3 |
| 4 (Fort) | .9 | .6 | 1.5 |
| 5 (C) | 210 | 150 | 1.4 |
| 5 (Fort) | 780 | 430 | 1.8 |

Table 2

Another group of tests were run in which four compute bound C programs and the equivalent Fortran programs were timed. These programs were taken from tests conducted to compare execution times on an IBM 370/168 and a DEC 11/45 [4]. A very small amount of actual arithmetic is done in each program, and there are no explicit subroutine calls. Programs 1 and 3 performed all computations using integer variables, and programs 2 and 4 (respectively) performed the same computations using floating point variables. All of the timing was done by using the "time" command on each program's execution. These results are summarized in Table 3. The system CPU times have been purposely omitted from Table 3, because in all cases they were either zero or negligible.

| Program | Real | User CPU | Ratio CPU |
|---|---|---|---|
| Config C2 (11/45) | | | |
| 1 (C) | 11.3 | 11.0 | |
| 1 (Fort) | 1:38.3 | 1:38.0 | |
| 2 (C) | 23.0 | 22.9 | |
| 2 (Fort) | 2:13.7 | 2:13.6 | |
| 3 (C) | 15.3 | 14.9 | |
| 3 (Fort) | 2:24.0 | 2:23.9 | |
| 4 (C) | 32.0 | 32.0 | |
| 4 (Fort) | 2:15.7 | 2:15.5 | |
| Config C7 (11/70) | | | |
| 1 (C) | 4.7 | 4.1 | 2.7 |
| 1 (Fort) | 45.0 | 41.9 | 2.3 |
| 2 (C) | 16.0 | 15.8 | 1.4 |
| 2 (Fort) | 1:07.0 | 1:05.4 | 2.0 |
| 3 (C) | 7.3 | 7.3 | 2.0 |
| 3 (Fort) | 1:24.3 | 1:23.5 | 1.7 |
| 4 (C) | 26.3 | 25.5 | 1.3 |
| 4 (Fort) | 1:15.7 | 1:14.2 | 1.8 |

Table 3

In order to get a feel for the effect of the cache itself, a program was written that was small enough to be run almost entirely from the cache. This program repetitively generated a random number that was used as an index to read a location from an integer array that encompassed the remainder of the program's address space. All arithmetic was done with integer variables, and a well-rounded mix of instructions was used in the program. All timing was done internally via the appropriate system calls. These test results are given in Table 4. System CPU times were either zero or negligible and have been omitted from the table.

| Config | Real | User CPU |
|---|---|---|
| C3 (11/45) | 1:12.0 | 1:12.8 |
| C7 (11/70) | 33.5 | 33.5 |

Table 4

The 11/45 configuration C3 is the same as C2 except that only fifteen (versus forty) I/O buffers are used. This was necessitated because there was insufficient memory to load this program when configuration C2 was used. There are two interesting results from this test. The ratio of 11/45 user CPU time to 11/70 user CPU time is 2.2, which is the same as that achieved by the chess procedure. But the size of the chess program is over ten times that of this program. The cache used on the 11/70 must, therefore, be classified as being very effective. Second, the prototype 11/70 had a "black box" attached to display the cache hit rate (in this case, the ratio of cache read hits to total memory reads and writes) as averaged over a million memory cycles. For this program the hit rate displayed was 85.3%.

2. I/O Bound Programs

Only one series of almost totally I/O bound timing experiments were run. Each test in this group was run single thread to eliminate the possibility of interference from swapping and contention for the few CPU cycles that were needed. All of these I/O tests on the 11/70 involved either the RP04 or the RS04 (RS03) on the high speed controller. Unfortunately, these devices were not available on the 11/45 being benchmarked, so an exact timing comparison was not possible. The mechanics of the tests involved timing, with the "time" command, "raw" I/O done by a "dd" command that read and wrote data in 8192 word blocks (ie. 32 256-word blocks). Table 5 summarizes the results of these tests. The CPU times given are the total CPU time required (user plus system).

| Devices | # blks | Real | CPU | µsec/wd |
|---|---|---|---|---|
| Config C2 (11/45) | | | | |
| RF11 to RP03 | 32 | 10.0 | .4 | 36.6 |
| RK05 to RP03 | 100 | 29.0 | .8 | 34.4 |
| RP03 to RP03* | 100 | 28.0 | .8 | 33.2 |
| RP03 to RP03 | 100 | 21.0 | .8 | 24.6 |
| Config C7 (11/70) | | | | |
| RP04 to RP04* | 100 | 13.0 | .5 | 15.2 |
| RS04 to RP04 | 64 | 4.0 | .4 | 6.8 |
| RS03 to RP04 | 32 | 3.0 | .2 | 10.6 |
| RP03 to RP04 | 100 | 14.0 | .5 | 16.4 |
| RK05 to RP04 | 100 | 22.0 | .5 | 26.2 |

Table 5

The µsec/word column in Table 5 gives the average data transfer time required for each word. It was computed by dividing real time less CPU time by the number of words of data transferred. An asterisk (*) in the Devices column indicates that the transfer was from the device to itself.

3. Mixed I/O and CPU

Timing CPU bound and I/O bound programs is certainly important and necessary in evaluating any system, but it is also prudent to evaluate performance using programs that use a mix of CPU and I/O. All of these tests used shell procedures that were run single thread so as to eliminate any possible contention from other processes. Each test procedure was timed using the "time" command. The first use of the memory device driver was made during these evaluations. This driver used 200K words of main memory to emulate an I/O device.

The first test consisted of timing the recompilation of a UNIX system. More specifically, a shell procedure was written to compile each of the system modules and archive them in the appropriate library, compile each of the device drivers and archive them in their library, compile the configuration table, assemble the two assembly language modules, and, finally, link edit everything together into a UNIX. Table 6 summarizes the results of these timing experiments. The CPU time is the total amount required (user plus system). There are two ratios expressed in the Ratio Real column. The first is the ratio of 11/45 configuration C1 real time to the 11/70 real time. The ratio in parentheses uses 11/45 configuration C2 real time in a like manner.

| Config | Real | CPU | Ratio Real |
|---|---|---|---|
| C1 | 17:39 | 6:54 | |
| C2 | 13:28 | 7:00.7 | |
| C5 | 8:34 | 3:29 | 2.0 (1.6) |
| C6 | 5:34 | 3:31.3 | 3.2 (2.4) |
| C7 | 5:19 | 3:31.4 | 3.3 (2.5) |
| C8 * | 5:06 | 3:43 | 3.5 (2.6) |

Table 6

The asterisk (*) by C8 indicates that two additional device drivers were included inadvertently in the compilations. Therefore, the times given for this configuration should be somewhat less and the ratios higher.

A similar test was then run in which the Fortran compiler was built. A shell procedure was used to assemble the various modules of this compiler and link edit them into an executable load module. It then assembled all of the Fortran builtin functions and I/O routines and placed them in their respective libraries. The results of this test are shown in Table 7. The CPU and Ratio Real columns of this table have the same meaning as in Table 6.

| Config | Real | CPU | Ratio Real |
|---|---|---|---|
| C1 | 14:41 | 3:14.7 | |
| C2 | 10:05 | 3:12 | |
| C7 | 4:01 | 1:46.6 | 3.7 (2.5) |
| C8 | 3:44 | 1:45.9 | 3.9 (2.7) |
| C9 | 3:15 | 1:54.2 | 4.5 (3.1) |

Table 7

The last test in this group used a program written to read into memory the file of encrypted login passwords (found in the file /etc/passwd), then read an input word, encrypt the word, and attempt to find a match among the passwords. This program was used twice in a shell procedure. The first occurrence simply accepted as input the dictionary word file /usr/dict/words (which contains over 72000 words). The second time it accepted as input the output of a program that read /usr/dict/words and reversed the letters of each word. The data was passed between the two programs via a pipe. These results are given in Table 8.

## 4. Load Testing

This last group of tests was designed to introduce and measure the effects of the resource contention that occurs when multiple

| Config | Real | CPU | Ratio Real |
|---|---|---|---|
| C1 | 12:43 | 12:26.5 | |
| C2 | 12:32 | 12:28.5 | |
| C4 | 5:24 | 5:08.6 | 2.4 |
| C7 | 5:12 | 5:10.0 | 2.4 |

Table 8

processes are active. The timing results give some indication of the response time (or, if you prefer, throughput) that may be expected when the system is put under stress.

To measure the effect of the 128K word memory on response time, a compute bound C program was written that required about 16K words of memory. The program itself was quite small and it performed trivial integer calculations on a large external array. All timing was done internally by the program. A single thread execution provided a response time to be used as a baseline. Then two, five, ten, and fifteen asynchronous executions of the program were done to introduce an artificial simulation of loading. Table 9 shows the results of these tests. The # Execs column shows how many simultaneous program executions were run, and the CPU column shows the total CPU time required (user plus system). The real time given in this table is the average of all the programs run simultaneously. The Inc column is the ratio of the multiple process average real time to the real time needed for a single program execution on that configuration. This ratio provides some insight to the affect of the memory on the effectiveness of multiprogramming. For example, running two processes simultaneously on the 11/45 took 2.5 times longer than the single thread execution. For improved throughput it would be better in this case not to multiprogram, because of the heavy response time price paid for interference. The Ratio Real column shows the ratio of 11/45 Real time to 11/70 Real time for that number of processes. For any particular line of data in the table, the times given are the average of all program executions in all test runs.

| # Execs | Real | CPU | Inc | Ratio Real |
|---|---|---|---|---|
| Config C2 (11/45) | | | | |
| 1 | 22.3 | 22.7 | | |
| 2 | 54.7 | 22.9 | 2.5 | |
| 5 | 2:26.8 | 23.0 | 6.6 | |
| 10 | 4:22.8 | 23.0 | 11.8 | |
| 15 | Ran Out Of Swap Space | | | |
| Config C7 (11/70) | | | | |
| 1 | 8.5 | 8.5 | | 2.6 |
| 2 | 15.8 | 8.5 | 1.9 | 3.5 |
| 5 | 39.4 | 8.6 | 4.6 | 3.7 |
| 10 | 1:02.0 | 8.7 | 7.3 | 4.2 |
| 15 | 1:26.8 | 8.8 | 10.2 | - |

Table 9

Although it is doubtful that any installation will ever experience a load like the one generated in the preceding test, the results are useful from the viewpoint of determining the affects of additional memory. To provide some semblance of a real life situation, a shell procedure was written to be used as a command script. The mix of UNIX commands in this script was based on shell accounting data from the UNIX Support Group's machine and Department 9152's machines [5]. The script approximates the command mix experienced by these UNIX systems and provides a reproducible load for timing purposes.

A single script was executed to obtain a timing point of reference. Then five scripts were started asynchronously at fifteen second intervals. This staggered starting was done to reduce the possibility of scripts getting into synchronization. Finally, nine scripts were started in a similar manner. This was the largest number of scripts that could be started without overflowing the process table. The "time" command was used to time the complete execution of each script and the shell accounting file provided timing data at the command level. Table 10 gives the times for complete scripts and Attachment C shows the distilled shell accounting information. The columns in Table 10 have the same meaning as those in Table 9.

| # Scripts | Real | CPU | Inc | Ratio Real |
|---|---|---|---|---|
| Config C2 (11/45) | | | | |
| 1 | 9:20.3 | 6:00.5 | | |
| 5 | 44:58.1 | 6:23.1 | 4.8 | |
| 9 | 87:49.3 | 6:38.9 | 9.4 | |
| Config C7 (11/70) | | | | |
| 1 | 4:18.7 | 3:13.3 | | 2.2 |
| 5 | 16:35.9 | 3:23.7 | 3.8 | 2.7 |
| 9 | 29:27.4 | 3:33.6 | 6.8 | 3.0 |
| Config C8 (11/70) | | | | |
| 1 | 4:07.0 | 3:11.2 | | 2.3 |
| 5 | 16:25.4 | 3:22.9 | 4.0 | 2.7 |

Table 10

The final experiment run was to measure the interference from heavy I/O activity to a CPU bound program. A subset of the chess shell procedure was chosen as the compute bound program to be timed. To induce heavy I/O, shell procedures were written to repetitively do "dd" commands. These "dd" commands would read from the device in a large block-size (between 5120 and 11264 words, depending on the device) and write the data to /dev/null (otherwise known as the bit bucket). This test was run on 11/70 configuration C7 only. Table 11 details the results of this test.

| I/O Devices | Usr CPU | Ratio Alone | wds/µsec |
|---|---|---|---|
| Pgm Alone | 2:11 | | 0 |
| +RS04 | 2:28 | .89 | .5 |
| +RS04+RP04 | 2:48 | .78 | .9 |
| +RS04+RP04+RP03 | 3:07 | .70 | 1.03 |
| +RS04+RP04+RP03+RK05 | 3:28 | .63 | 1.12 |

Table 11

The real and system CPU times have been omitted from Table 11 because the only interest was in the increase in the program's CPU requirements. The wds/µsec column shows how many words of data were being read each µsec, and the Ratio Alone column shows the ratio of the CPU time required for the program by itself to that required with the I/O interference.

Acknowledgements

We would like to express our appreciation to Mr. Steve Brown of DEC for his liaison work in setting up this benchmark expedition. Mr. D. W. Smith provided data that was of considerable assistance when writing the command scripts. Mr. T. M. Raleigh participated by supplying the RP04 driver and helping write and debug the RS04 and TU16 drivers.

R. B. Brandt

K. Thompson

MH-8234-RBB-nroff
KT

Atts.
Attachments A-C
Figure 1

# REFERENCES


[1] PDP 11/70 Processor Handbook, Digital Equipment Corporation.

[2] "UNIX Driver for DEC RP04 Disk", forthcoming memorandum by T. M. Raleigh.

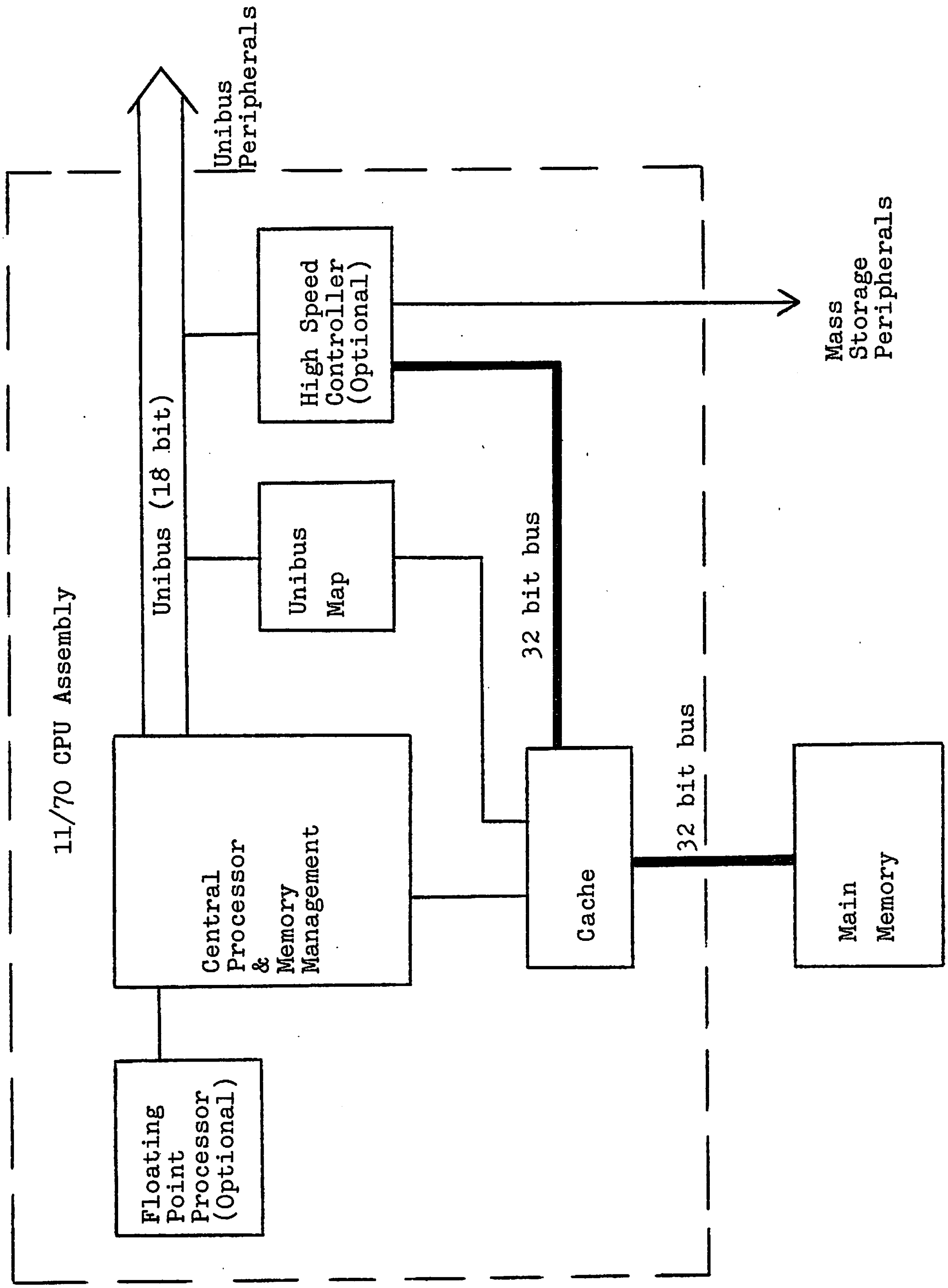
[3] "The Gibson Mix", J. C. Gibson, IBM Technical Report TR00.2043, June 18, 1970.

[4] "Performance Tests on a DEC PDP 11/45 Mini-computer - File 38730-039", J. E. Cloutier, March 25, 1975.

[5] "UNIX Command Usage", letter from D. W. Smith to R. B. Brandt, January 15, 1975.

FIGURE 1

11/7Ø Architecture



ATTACHMENT A

## Hardware Available on Benchmark 11/70

### 11/70 Processor Assembly

11/70 Processor with Cache Memory
Floating Point Processor FP70
Memory Management Unit MM70
Unibus Map
Four High Speed Controllers RH70
Line Clock KW11L
Programmable Clock KW11P

### Attached To High Speed Controllers

RP04 Moving Head Disk
RS03 and RS04 Fixed Head Disk
TU16 Magnetic Tape Drive
DEC Diagnostic Equipment

### Memory

640K words of Main Memory

### Attached To Unibus

DECwriter (console) on DL11A
RP03 Moving Head Disk
DECtape TC11
DH11 Programmable Asynchronous Multiplexer
2 RK05 Moving Head Disks
Paper Tape Reader/Punch

## Hardware Available on Benchmark 11/45

### 11/45 Processor Assembly

11/45 Processor
Floating Point Processor
Memory Management Unit
Line Clock
Programmable Clock

### Memory

48K words of Main Memory

### Attached To Unibus

DECwriter (console)
2 RP03 Moving Head Disk
DECtape TC11
DHDM Programmable Asynchronous Multiplexer
2 RK05 Moving Head Disks
Paper Tape Reader/Punch
Line Printer
RF11 Fixed Head Disk
TU10 Magnetic Tape Drive

# ATTACHMENT B

## Benchmark System Configurations

| Config | CPU | Root | TMP | Swap | USR | I/O Buffs |
|---|---|---|---|---|---|---|
| C1 | 11/45 | RK05 | RK05 | RK05 | RP03 | 15 |
| C2 | 11/45 | RK05 | RK05 | RF11 | RP03 | 40 |
| C3 | 11/45 | RK05 | RK05 | RF11 | RP03 | 15 |
| C4 | 11/70 | RK05 | RK05 | RK05 | RP03 | 40 |
| C5 | 11/70 | RK05 | RS04 | RK05 | RP03 | 40 |
| C6 | 11/70 | RS04 | RS04 | RS03 | RP03 | 40 |
| C7 | 11/70 | RS04 | RS04 | RS03 | RP04 | 40 |
| C8 | 11/70 | RS04 | Memory | RS03 | RP04 | 40 |
| C9 | 11/70 | Memory | Memory | RS03 | RP04 | 40 |

ATTACHMENT C

## Command Script Timing Results

### 11/45 Configuration C2, 1 Script
### Summation of 3 Runs

| Command | No. Cmds | | Avg Real(secs) | | Avg CPU(secs) | | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 54 | 17.31% | 9.04 | 19.00% | 6.51 | 20.37% | 1.4 |
| sh | 36 | 11.54% | 26.53 | 37.17% | 18.66 | 38.93% | 1.4 |
| ls | 36 | 11.54% | 2.44 | 3.43% | 1.30 | 2.71% | 1.9 |
| cat | 30 | 9.62% | 1.93 | 2.26% | 0.93 | 1.61% | 2.1 |
| rm | 24 | 7.69% | 1.42 | 1.32% | 0.40 | 0.56% | 3.5 |
| cp | 18 | 5.77% | 1.39 | 0.97% | 0.37 | 0.38% | 3.8 |
| cc | 12 | 3.85% | 35.25 | 16.47% | 20.31 | 14.12% | 1.7 |
| pr | 12 | 3.85% | 6.67 | 3.11% | 5.73 | 3.99% | 1.2 |
| du | 12 | 3.85% | 1.17 | 0.54% | 0.64 | 0.44% | 1.8 |
| mv | 12 | 3.85% | 0.50 | 0.23% | 0.15 | 0.10% | 3.4 |
| **gok | 6 | 1.92% | 55.33 | 12.92% | 29.19 | 10.15% | 1.9 |
| colv | 6 | 1.92% | 1.00 | 0.23% | 13.71 | 4.77% | 0.1 |
| ps | 6 | 1.92% | 4.00 | 0.93% | 2.18 | 0.76% | 1.8 |
| grep | 6 | 1.92% | 3.00 | 0.70% | 2.13 | 0.74% | 1.4 |
| pwd | 6 | 1.92% | 1.17 | 0.27% | 0.38 | 0.13% | 3.0 |
| rmdir | 6 | 1.92% | 0.33 | 0.08% | 0.17 | 0.06% | 2.0 |
| who | 6 | 1.92% | 0.17 | 0.04% | 0.12 | 0.04% | 1.4 |
| echo | 6 | 1.92% | 0.17 | 0.04% | 0.12 | 0.04% | 1.4 |
| mkdir | 6 | 1.92% | 0.33 | 0.08% | 0.11 | 0.04% | 3.0 |
| date | 6 | 1.92% | 0.50 | 0.12% | 0.09 | 0.03% | 5.6 |
| chmod | 6 | 1.92% | 0.33 | 0.08% | 0.08 | 0.03% | 4.3 |
| Total | 312 | | 8.23 | | 5.53 | | 1.5 |

Note: Because of its use with a pipe, the "nroff" command used in the script was recorded in the shell accounting file as "**gok".

11/45 Configuration C2, 5 Scripts
Summation of 3 Runs

| Command | No. | Cmds | Avg Real(secs) | | Avg CPU(secs) | | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 260 | 16.95% | 40.73 | 17.85% | 7.05 | 20.44% | 5.8 |
| sh | 182 | 11.86% | 117.82 | 36.15% | 19.44 | 39.46% | 6.1 |
| ls | 174 | 11.34% | 15.02 | 4.40% | 1.60 | 3.11% | 9.4 |
| cat | 148 | 9.65% | 7.55 | 1.88% | 0.95 | 1.57% | 7.9 |
| rm | 118 | 7.69% | 7.49 | 1.49% | 0.43 | 0.56% | 17.5 |
| cp | 90 | 5.87% | 4.52 | 0.69% | 0.37 | 0.37% | 12.3 |
| pr | 60 | 3.91% | 25.48 | 2.58% | 5.83 | 3.90% | 4.4 |
| du | 59 | 3.85% | 10.22 | 1.02% | 1.24 | 0.82% | 8.2 |
| mv | 58 | 3.78% | 5.36 | 0.52% | 0.14 | 0.09% | 37.9 |
| cc | 57 | 3.72% | 196.25 | 18.86% | 21.36 | 13.58% | 9.2 |
| colv | 30 | 1.96% | 5.17 | 0.26% | 13.86 | 4.64% | 0.4 |
| ps | 30 | 1.96% | 24.67 | 1.25% | 2.37 | 0.79% | 10.4 |
| pwd | 30 | 1.96% | 6.07 | 0.31% | 0.40 | 0.13% | 15.1 |
| rmdir | 30 | 1.96% | 2.33 | 0.12% | 0.16 | 0.05% | 14.5 |
| mkdir | 30 | 1.96% | 2.80 | 0.14% | 0.13 | 0.04% | 20.8 |
| who | 30 | 1.96% | 4.33 | 0.22% | 0.11 | 0.04% | 40.6 |
| date | 30 | 1.96% | 2.03 | 0.10% | 0.10 | 0.03% | 20.0 |
| chmod | 30 | 1.96% | 3.00 | 0.15% | 0.08 | 0.03% | 39.7 |
| **gok | 29 | 1.89% | 223.93 | 10.95% | 29.69 | 9.60% | 7.5 |
| grep | 29 | 1.89% | 18.34 | 0.90% | 2.19 | 0.71% | 8.4 |
| echo | 29 | 1.89% | 3.41 | 0.17% | 0.11 | 0.03% | 31.9 |
| Total | 1534 | | 38.67 | | 5.85 | | 6.6 |

11/45 Configuration C2, 9 Scripts
Summation of 2 Runs

| Command | No. Cmds | | Avg Real(secs) | | Avg CPU(secs) | | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 302 | 16.68% | 83.22 | 18.38% | 7.14 | 19.44% | 11.7 |
| sh | 218 | 12.04% | 224.56 | 35.81% | 20.32 | 39.93% | 11.1 |
| ls | 206 | 11.37% | 32.06 | 4.83% | 1.89 | 3.51% | 16.9 |
| cat | 177 | 9.77% | 16.76 | 2.17% | 0.98 | 1.56% | 17.1 |
| rm | 138 | 7.62% | 17.01 | 1.72% | 0.44 | 0.55% | 38.3 |
| cp | 103 | 5.69% | 10.12 | 0.76% | 0.38 | 0.35% | 26.8 |
| mv | 71 | 3.92% | 10.51 | 0.55% | 0.15 | 0.09% | 71.8 |
| du | 70 | 3.87% | 29.53 | 1.51% | 1.94 | 1.23% | 15.2 |
| cc | 69 | 3.81% | 368.81 | 18.62% | 21.14 | 13.15% | 17.4 |
| pr | 68 | 3.75% | 47.34 | 2.35% | 5.45 | 3.34% | 8.7 |
| **gok | 36 | 1.99% | 340.03 | 8.96% | 29.78 | 9.66% | 11.4 |
| colv | 36 | 1.99% | 8.22 | 0.22% | 13.95 | 4.53% | 0.6 |
| pwd | 36 | 1.99% | 11.61 | 0.31% | 0.39 | 0.13% | 29.8 |
| who | 36 | 1.99% | 8.03 | 0.21% | 0.14 | 0.05% | 55.6 |
| mkdir | 36 | 1.99% | 6.83 | 0.18% | 0.14 | 0.04% | 50.0 |
| date | 36 | 1.99% | 4.64 | 0.12% | 0.10 | 0.03% | 44.7 |
| ps | 35 | 1.93% | 76.86 | 1.97% | 5.10 | 1.61% | 15.1 |
| rmdir | 35 | 1.93% | 6.00 | 0.15% | 0.17 | 0.06% | 34.3 |
| echo | 35 | 1.93% | 5.86 | 0.15% | 0.11 | 0.04% | 51.3 |
| grep | 34 | 1.88% | 34.56 | 0.86% | 2.22 | 0.68% | 15.6 |
| chmod | 34 | 1.88% | 6.76 | 0.17% | 0.08 | 0.03% | 82.1 |
| Total | 1811 | | 75.48 | | 6.13 | | 12.3 |

11/70 Configuration C7, 1 Script
Summation of 3 Runs

| Command | No. Cmds | | Avg Real(secs) | | Avg CPU(secs) | | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 54 | 17.25% | 4.67 | 21.45% | 3.86 | 22.67% | 1.2 |
| sh | 36 | 11.50% | 12.14 | 37.19% | 9.91 | 38.84% | 1.2 |
| ls | 36 | 11.50% | 1.03 | 3.15% | 0.67 | 2.63% | 1.5 |
| cat | 31 | 9.90% | 1.26 | 3.32% | 0.50 | 1.69% | 2.5 |
| rm | 24 | 7.67% | 0.50 | 1.02% | 0.22 | 0.57% | 2.3 |
| cp | 18 | 5.75% | 0.67 | 1.02% | 0.23 | 0.44% | 3.0 |
| cc | 12 | 3.83% | 14.75 | 15.06% | 10.70 | 13.98% | 1.4 |
| pr | 12 | 3.83% | 3.17 | 3.23% | 2.64 | 3.45% | 1.2 |
| du | 12 | 3.83% | 0.67 | 0.68% | 0.36 | 0.47% | 1.8 |
| mv | 12 | 3.83% | 0.33 | 0.34% | 0.08 | 0.11% | 4.1 |
| **gok | 6 | 1.92% | 20.00 | 10.21% | 12.99 | 8.49% | 1.5 |
| colv | 6 | 1.92% | 0.00 | 0.00% | 6.29 | 4.11% | 0.0 |
| ps | 6 | 1.92% | 2.67 | 1.36% | 2.30 | 1.50% | 1.2 |
| grep | 6 | 1.92% | 2.00 | 1.02% | 0.98 | 0.64% | 2.0 |
| pwd | 6 | 1.92% | 0.83 | 0.43% | 0.21 | 0.14% | 4.0 |
| mkdir | 6 | 1.92% | 0.33 | 0.17% | 0.09 | 0.06% | 3.8 |
| rmdir | 6 | 1.92% | 0.00 | 0.00% | 0.09 | 0.06% | 0.0 |
| who | 6 | 1.92% | 0.17 | 0.09% | 0.07 | 0.05% | 2.3 |
| echo | 6 | 1.92% | 0.17 | 0.09% | 0.06 | 0.04% | 2.7 |
| date | 6 | 1.92% | 0.17 | 0.09% | 0.06 | 0.04% | 3.0 |
| chmod | 6 | 1.92% | 0.17 | 0.09% | 0.04 | 0.03% | 3.8 |
| Total | 313 | | 3.75 | | 2.93 | | 1.3 |

11/70 Configuration C7, 5 Scripts
Summation of 3 Runs

| Command | No. | Cmds | Avg | Real(secs) | Avg | CPU(secs) | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 266 | 17.43% | 22.08 | 26.67% | 4.18 | 23.37% | 5.3 |
| sh | 183 | 11.99% | 45.13 | 37.51% | 10.20 | 39.22% | 4.4 |
| ls | 174 | 11.40% | 4.56 | 3.61% | 0.81 | 2.96% | 5.6 |
| cat | 148 | 9.70% | 3.16 | 2.13% | 0.53 | 1.65% | 5.9 |
| rm | 114 | 7.47% | 2.12 | 1.10% | 0.22 | 0.54% | 9.5 |
| cp | 87 | 5.70% | 1.80 | 0.71% | 0.21 | 0.39% | 8.4 |
| pr | 59 | 3.87% | 7.80 | 2.09% | 2.61 | 3.23% | 3.0 |
| cc | 58 | 3.80% | 52.50 | 13.83% | 11.00 | 13.40% | 4.8 |
| du | 58 | 3.80% | 3.81 | 1.00% | 0.65 | 0.79% | 5.9 |
| mv | 56 | 3.67% | 0.68 | 0.17% | 0.08 | 0.09% | 9.0 |
| colv | 30 | 1.97% | 0.27 | 0.04% | 6.37 | 4.01% | 0.0 |
| ps | 30 | 1.97% | 21.43 | 2.92% | 2.48 | 1.56% | 8.6 |
| pwd | 30 | 1.97% | 2.67 | 0.36% | 0.21 | 0.13% | 12.8 |
| mkdir | 30 | 1.97% | 0.70 | 0.10% | 0.09 | 0.06% | 7.9 |
| who | 30 | 1.97% | 0.93 | 0.13% | 0.09 | 0.05% | 10.8 |
| date | 30 | 1.97% | 0.33 | 0.05% | 0.06 | 0.04% | 5.9 |
| grep | 29 | 1.90% | 4.59 | 0.60% | 1.01 | 0.62% | 4.5 |
| rmdir | 29 | 1.90% | 0.76 | 0.10% | 0.10 | 0.06% | 7.6 |
| chmod | 29 | 1.90% | 0.45 | 0.06% | 0.04 | 0.03% | 10.4 |
| **gok | 28 | 1.83% | 53.25 | 6.77% | 13.20 | 7.77% | 4.0 |
| echo | 28 | 1.83% | 0.57 | 0.07% | 0.06 | 0.04% | 9.2 |
| Total | 1526 | | 14.43 | | 3.12 | | 4.6 |

## 11/70 Configuration C7, 9 Scripts
## Summation of 2 Runs

| Command | No. Cmds | | Avg Real(secs) | | Avg CPU(secs) | | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 309 | 16.94% | 36.91 | 22.67% | 4.26 | 21.92% | 8.7 |
| sh | 218 | 11.95% | 94.33 | 40.87% | 11.00 | 39.95% | 8.6 |
| ls | 211 | 11.57% | 6.66 | 2.79% | 0.92 | 3.23% | 7.2 |
| cat | 173 | 9.48% | 3.25 | 1.12% | 0.55 | 1.58% | 5.9 |
| rm | 139 | 7.62% | 2.60 | 0.72% | 0.23 | 0.54% | 11.1 |
| cp | 106 | 5.81% | 1.96 | 0.41% | 0.22 | 0.39% | 8.9 |
| mv | 71 | 3.89% | 0.82 | 0.12% | 0.08 | 0.10% | 10.1 |
| cc | 70 | 3.84% | 79.03 | 10.99% | 10.99 | 12.81% | 7.2 |
| du | 70 | 3.84% | 4.67 | 0.65% | 0.81 | 0.94% | 5.8 |
| pr | 68 | 3.73% | 9.90 | 1.34% | 2.26 | 2.56% | 4.4 |
| rmdir | 36 | 1.97% | 1.00 | 0.07% | 0.10 | 0.06% | 10.4 |
| date | 36 | 1.97% | 0.39 | 0.03% | 0.06 | 0.04% | 6.6 |
| chmod | 36 | 1.97% | 0.50 | 0.04% | 0.05 | 0.03% | 10.0 |
| **gok | 35 | 1.92% | 174.26 | 12.12% | 13.24 | 7.72% | 13.2 |
| colv | 35 | 1.92% | 0.40 | 0.03% | 6.42 | 3.74% | 0.1 |
| ps | 35 | 1.92% | 76.31 | 5.31% | 6.04 | 3.52% | 12.6 |
| grep | 35 | 1.92% | 4.57 | 0.32% | 1.03 | 0.60% | 4.4 |
| pwd | 35 | 1.92% | 3.09 | 0.21% | 0.24 | 0.14% | 12.9 |
| who | 35 | 1.92% | 1.11 | 0.08% | 0.08 | 0.05% | 13.8 |
| mkdir | 35 | 1.92% | 0.71 | 0.05% | 0.08 | 0.05% | 9.1 |
| echo | 35 | 1.92% | 0.94 | 0.07% | 0.06 | 0.03% | 16.2 |
| Total | 1824 | | 27.58 | | 3.29 | | 8.4 |

# 11/70 Configuration C8, 5 Scripts
## Summation of 1 Run

| Command | No. Cmds | | Avg Real(secs) | | Avg CPU(secs) | | Real/CPU |
|---|---|---|---|---|---|---|---|
| ed | 89 | 17.32% | 25.39 | 30.43% | 4.27 | 24.16% | 5.9 |
| sh | 61 | 11.87% | 47.43 | 38.95% | 10.10 | 39.17% | 4.7 |
| ls | 59 | 11.48% | 4.22 | 3.35% | 0.77 | 2.88% | 5.5 |
| cat | 49 | 9.53% | 2.69 | 1.78% | 0.53 | 1.66% | 5.1 |
| rm | 39 | 7.59% | 1.82 | 0.96% | 0.21 | 0.52% | 8.6 |
| cp | 30 | 5.84% | 1.63 | 0.66% | 0.22 | 0.42% | 7.3 |
| pr | 20 | 3.89% | 7.15 | 1.93% | 2.68 | 3.41% | 2.7 |
| du | 20 | 3.89% | 3.80 | 1.02% | 0.61 | 0.78% | 6.2 |
| mv | 19 | 3.70% | 0.74 | 0.19% | 0.07 | 0.09% | 10.1 |
| cc | 18 | 3.50% | 45.17 | 10.95% | 11.12 | 12.73% | 4.1 |
| **gok | 10 | 1.95% | 51.30 | 6.91% | 13.20 | 8.39% | 3.9 |
| colv | 10 | 1.95% | 0.20 | 0.03% | 6.30 | 4.01% | 0.0 |
| ps | 10 | 1.95% | 10.90 | 1.47% | 1.17 | 0.74% | 9.3 |
| grep | 10 | 1.95% | 4.10 | 0.55% | 1.02 | 0.65% | 4.0 |
| pwd | 10 | 1.95% | 2.50 | 0.34% | 0.22 | 0.14% | 11.6 |
| rmdir | 10 | 1.95% | 0.80 | 0.11% | 0.09 | 0.06% | 8.7 |
| mkdir | 10 | 1.95% | 0.80 | 0.11% | 0.09 | 0.06% | 8.7 |
| who | 10 | 1.95% | 0.80 | 0.11% | 0.08 | 0.05% | 10.2 |
| date | 10 | 1.95% | 0.30 | 0.04% | 0.05 | 0.03% | 6.4 |
| chmod | 10 | 1.95% | 0.60 | 0.08% | 0.04 | 0.03% | 13.8 |
| echo | 9 | 1.75% | 0.44 | 0.05% | 0.06 | 0.03% | 7.5 |
| Total | 514 | | 14.45 | | 3.06 | | 4.7 |