SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED INTERACTIVE COMPUTING NOTE 808

PERQ UNIX Implementation Note # 60
Benchmark results for PNX Release 1

Issued by
James Collis

22 February 1983

DISTRIBUTION:
R W Witty
K Robinson
C Prosser
A S Williams
L O Ford
I D Benest
T Watson
E V C Fielding
J C Malone
P J Smith
A J Kinroy
C P Wadsworth
J R Collis
M Ritchie ICL Dalkeith
P Palmer ICL Dalkeith
RL Support/PERQ/Unix Implementation Notes File
CBP PERQ Evaluation and Benchmarking

1. Introduction

This note compares the results of benchmarking PNX Release 1 on a half MB PERQ, with the results already obtained for :-

PDP 11/70
PERQ running Accent Unix (1 MB)
PERQ running early Microcode Unix (half MB)

in PERQ UNIX Implementation Note 40. The earlier results are reproduced with the results for PNX.

2. Summary of results.

Though individual results vary, in general the PNX Release 1 benchmark results are very similar to the earlier Microcode UNIX, which is as one would expect. Overall PNX is slightly slower, writing to disc in particular (see Kashtan), while some system calls are much faster. Noting that the PDP 11/70 results were obtained while only one person was using the machine, the PNX results compare favourably.

PDP sp $

3. C Instructions

Times for various instructions in C using single precision. All times are in microseconds unless otherwise specified.

Single Precision (16 bits)				
Instruction	AccUnix PERQ	Mi-Unix PERQ	PNX PERQ	PDP 11/70
FOR loop (1)	195	239	256	48
REPEAT loop (1)	157	220	237	48
WHILE loop (1)	197	239	255	55
x = 0;	2.3	6	6	2.4
x = a;	3.7	11	12	2.4
x = a/2*3+4-5;	41	35	36	14
x = a/a*a+a-a;	52	60	63	16
ar[j] = ar[j-1];	22	22	25	6
IF (x<50) ;ELSE ;	13	11	13	1.6
IF (x<1) ;ELSE ;	13	9	10	1.4
5 nested proc calls	484	78	78	110
with value par's	590	120	126	117
with ref par's	522	99	102	114
x = a + b;	14	15	18	4
x = a - b;	14	15	18	4
x = a * b;	24	24	26	7
x = a / b;	33	32	34	10
x = a % b;	33	32	34	10
x = *p;	5	8	8	3
*p++;	11	6	6	4

(1) 10 iterations of each loop.

The PNX results are slightly slower all round than the earlier version, by on average between 5% and 10%. These 16 bit results compare poorly with the PDP 11/70, but are not so important because the PERQ is a 32 bit machine.

Times for various instructions in C using double precision. All times are in microseconds unless otherwise specified.

Double Precision (32 bits)				
Instruction	AccUnix PERQ	Mi-Unix PERQ	PNX PERQ	PDP 11/70
FOR loop (1)	232	155	158	102
REPEAT loop (1)	194	147	155	122
WHILE loop (1)	222	156	157	89
x = 0;	1.9	1.4	2.1	5.3
x = a;	4.2	2.1	2.8	5.4
x = a/2*3+4-5;	36	28	28	97
x = a/a*a+a-a;	44	36	37	102
ar[j] = ar[j-1];	40	22	22	11
IF (x<50) ;ELSE ;	14	7.6	7.4	2.3
IF (x<1) ;ELSE ;	12	5	4.2	2.1
5 nested proc calls	484	78	78	110
with value par's	537	103	104	130
with ref par's	504	102	102	132
x = a + b;	9	4	4	8
x = a - b;	9	4	4	8
x = a * b;	19	13	13	46
x = a / b;	28	21	21	55
x = a % b;	28	21	21	56
x = *p;	10	4	4.9	6
*p++;	11	6	6	5
x = chararray[k];	24	20	23	4
x = a[k].ifield;	23	14	15	9
1 empty proc call	72	15	15	22
x = getpid();	70ms	223us	243us	370us
stat(buf,&statbuf);	208ms	24ms	19ms	8ms
Open & Close a file	>1sec	26ms	19ms	7ms

(1) 10 iterations of each loop.

These results again are very similar to before, the exceptions being the improvement in the last two system calls for file statistics (stat), and opening and closing a file. The PNX results compare well with the PDP, especially as all PDP benchmarks were run while only one person was using the machine, and so give a favourable impression of the 11/70 which is a multi-user machine.

4. Kashtan Programs

Programs kt5 and kt6 are the only ones to highlight any differences between the early Microcode and PNX.

kt5 Writes 2 MBytes to disk.
kt6 Randomly positions within a disk file, writes half KByte on to disk and repeats 4000 times.
kt7 Two processes signal each other 20,000 times.
kt8 One process signalling itself 1,000 times.
kt9 Writes 5 MBytes to itself through a pipe.
kt10 Transmits 5 MBytes between two processes through a pipe.
kt11 Two processes each transmitting 5 MBytes to each other.

All times in minutes and seconds.

	PDP 11/70			Microcode Unix			PNX Release 1		
Name	Real	User	Sys	Real	User	Sys	Real	User	Sys
kt5	1:42	0.1	22.1	45	0.1	27.8	2:40	0.2	28.3
kt6	30	0.7	10.4	7:53	0.2	3:58	9:41	0.1	5:15.8
kt7	1:24	1.3	25.9	1:10	2.2	32.4	1:14	0.7	31.8
kt8	5	0.0	2.5	5	0.0	4.1	5	0.0	4.0
kt9	1:00	0.4	47.7	32	0.3	30.7	34	1.1	33.1
kt10	1:06	0.3	26.2	42	0.7	19.2	46	0.5	20.1
kt11	2:15	0.7	53.9	1:24	0.6	50.8	1:30	0.6	40.8

5. Zilog Programs

What each program does:

Sieve	20 iterations of sieve on 0 - 8191.
Disk_test	a) 1000 sequential writes.
	b) 1000 sequential reads.
	c) 1000 random reads.
Proc_call	1,000,000 function calls to dummy(a,b,c) where dummy returns (a + b + c).
System_call	10,000 calls to "getpid".
Piper	2,000 pipe writes of 512 bytes. Total of 1,024,000 bytes.

Result times in seconds.

	sieve	disk test			proc	sys	piper
		a	b	c			
PDP 11/70	3.7	32.2	17.5	37.4	36.1	3.1	16.2
Acc Unix	27.0				148.4	712	398
Microcode	11.5	42.8	36.3	59.7	41.9	2.3	7.7
PNX	14.0	44.4	36.0	71.0	42.2	2.5	8.5

PDP.sp $