Lab 1 Report

Roll Number: 130050014, 130050031

To create files of fixed (2MB) size command used is:

dd if=/dev/zero of=foo0.txt bs=2000000 count=1

Ex 1:

Command: cat /proc/cpuinfo

Cpu cores: 4

Command: cat /proc/meminfo

MemTotal: 8142244 kB

MemFree: 2653988 kB

Therefore about **0.325** of memory is free.

Command: cat /proc/stat

ctxt 1888505977 **(context switches since booting)**

processes 1706078 **(number of processes forked since booting)**

Ex 2:

The set of commands used to monitor resources used by the process is common to all the five Programs. Just take the relevant data according to the process id (ps -aux).

Command: top

Monitored Resource: CPU utilization (%CPU) and Memory utilization (%MEM)

Command: iostat -d 2 -x

Monitored Resource: Disk utilization (%util) and I/O statistics (r/s: read, w/s: write)

Command: ifconfig eth0 (Note: take care other processes are not using network)

Monitored Resource: Network usage on eth0

The output for network usage is the same for all the five programs.

Its relevant part is as follows:

Output:

eth0 Link encap:Ethernet HWaddr 54:53:ed:ac:f7:85

RX packets:752132 errors:1 dropped:0 overruns:0 frame:1

TX packets:437769 errors:0 dropped:2 overruns:0 carrier:0

After few seconds,

eth0 Link encap:Ethernet HWaddr 54:53:ed:ac:f7:85

RX packets:752132 errors:1 dropped:0 overruns:0 frame:1

TX packets:437769 errors:0 dropped:2 overruns:0 carrier:0

No change in number of RX and TX packets!

A] cpu1.c:

Top Output:

PID USER %CPU %MEM TIME+ COMMAND

32346 palakja+ 99.8 0.0 22:32.35 cpu1

Iostat Output:

Device: r/s w/s %util

sda 0.00 0.00 0.00

dm-0 0.00 0.00 0.00

Bottleneck: CPU

Reason: Disk usage and Memory usage is negligible as the program is devoid of any I/O.

Therefore CPU is the bottleneck as the utilization is around 100%.

B] cpu1print.c:

Top Output:

PID USER %CPU %MEM TIME+ COMMAND

2069 palakja+ 6.0 0.0 0:07.95 cpu1print

5324palakja+ 99.4 0.9 10:07.95 **gnome-terminal-**

Iostat Output:

Device: r/s w/s %util

sda 0.00 11.00 0.00

dm-0 0.00 14.00 0.00

Bottleneck: CPU

Reason: No disk access done by the program also memory usage is negligible from the data depicted above. The **genome terminal process** uses CPU to its full capacity to print output to the screen, as this process caters to the system call made by "printf" function from the cpu1print.c . Therefore considering all other resources except CPU, we can conclude that CPU is bottleneck.

C] cpu2.c

Top Output:

PID USER %CPU %MEM TIME+ COMMAND

7058 palakja+ 100.0 0.0 1:03.91 cpu2

Iostat Output:

Device: r/s w/s %util

sda 0.00 0.00 0.00

dm-0 0.00 0.00 0.00

Bottleneck: CPU

Reason: Disk usage and Memory usage is negligible as the program is devoid of any I/O.

Therefore CPU is the bottleneck as the utilization is around 100%.

D] disk.c

Top Output:

PID USER %CPU %MEM TIME+ COMMAND

15249 rawalkh+ 17.0 0.0 0:42.58 disk

Iostat Output:

Device: r/s w/s %util

sda 2015.00 0.00 96.40

dm-0 2015.00 0.00 96.00

Bottleneck: Disk

Reason: The program is randomly reading files,as a result they cannot be read from the buffer readily.This results in very high number of disk access. The %util field is approximately 97% which denotes disk usage. Therefore disk is the bottleneck.

E] disk1.c

Top Output:

PID USER %CPU %MEM TIME+ COMMAND

15585 rawalkh+ 97.4 0.0 0:12.97 disk1

Iostat Output:

Device: r/s w/s %util

sda 14.00 5.00 1.20

dm-0 12.00 19.00 1.20

Bottleneck: CPU

Reason: The program reads files sequentially,as a result most of the data required is stored in the buffer (not many disk accesses). Also from the "top" output it is evident that the CPU usage is the bottleneck as it is nearly 97%.

Ex 3:

Command: cat /proc/<pid>/stat|awk '{print $14,$15}'

$14 :user mode time **$15: kernel mode time**

For each program following Data points are at two instances during runtime of program:

A] cpu1.c

<User mode time> <Kernel mode time>

2760 4

5878 9

Reason: No input/output required by the program. No system calls.

Therefore, the process mostly stays in user mode. Thus, **user mode time dominates.**

B] cpu1print.c

<User mode time> <Kernel mode time>

24 149

25 159

Reason: cpu1print prints to default output. System call made by "printf" function.

Hence process switches to kernel mode most of the time.

Therefore, **kernel mode time dominates.**

C] cpu2.c

<User mode time> <Kernel mode time>

2257 3

3478 5

Reason: No I/O to terminal. "gettimeoftheday" function does system calls to get time.

But such calls require very less system time. Thus, **user mode time dominates.**

Ex 4:

Note: Observation 1 and Observation 2 are made in few seconds of each other.

A] cpu1.c

Obsv 1: voluntary_ctxt_switches: 3

nonvoluntary_ctxt_switches: 1646

Obsv2: voluntary_ctxt_switches: 3

nonvoluntary_ctxt_switches: 4200

Reason: The program has no input/output requirements => no system calls.

As a result the context switches only occur mostly when OS scheduler

switches to other process. These is called as "**Timer Interrupt**".

Therefore **non-voluntary context switches are more**.

B] disk.c

Obsv 1: voluntary_ctxt_switches: 136537

nonvoluntary_ctxt_switches: 5070

Obsv2: voluntary_ctxt_switches: 140281

nonvoluntary_ctxt_switches: 5216

Reason: Program disk1.c reads from text files as a result it requests system call

to read. This is a **"blocking system call".** Instead of waiting for the disk, other processes

are executed, hence a context switch occurs. The process did this voluntarily.

Therefore, **voluntary context switches dominate.**