

## Operating Systems - CS 304

Balsher Singh - 180010008

### Assignment - 1

Q1.

- a.) After running 'more /proc/cpuinfo' command, info about cpu was displayed.

```
sher@DELL-G3:~$ more /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model        : 158
model name    : Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz
stepping     : 10
microcode    : 0xffffffff
cpu MHz      : 2304.000
cache size   : 256 KB
physical id  : 0
siblings     : 8
core id      : 0
cpu cores    : 4
apicid       : 0
initial apicid : 0
fpu          : yes
fpu_exception : yes
cpuid level  : 6
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat p
se36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm pni
pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 fma cx16 xtpr pdcm pcid sse4_1 sse
4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave osxsave avx f16c rdrand lahf_lm
abm 3dnowprefetch fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx
smap clflushopt intel_pt ibrs ibpb stibp ssbd
bogomips     : 4608.00
clflush size : 64
cache_alignm : 64
address sizes : 36 bits physical, 48 bits virtual
power management:
```

**Processor:** Provides each processor with an identifying number. If you have one processor it will display a 0. If you have more than one processor it will display all processor information separately counting the processors using zero notation.

**Cores:** No. of cores in that particular processor are shown in this info.

- b.) The following output is shown when lscpu is used:

Core(s) per socket: 4

Socket(s): 1

Total cores are (Core(s) per socket) × Socket(s) = 4\*1 = 4 cores.

- c.) After running the command 'more /proc/cpuinfo', I got that there are 8 logical processors in my system.

Can also be verified by 'cat /proc/cpuinfo | grep processor | wc -l' this command

- d.) Each processor has frequency of 2304 MHz as shown in 'cat /proc/cpuinfo'

- e.) This can be found by 'more /proc/meminfo' command.

```
sher@DELL-G3:~$ more /proc/meminfo
MemTotal:      16634424 kB
MemFree:       9818320 kB
```

It is 16634424kB for my machine.

- f.) As shown above, free memory is 9818320 kB.

g.) This command can be used to find this → 'vmstat -f'

[vmstat man page](#)

Output: 725 forks

Also, can be found using 'more /proc/stat' command. (725 forks shown below)

h.) Context Switches since bootup information can be received using the command:

'more /proc/stat': ctxt 494453

```
sher@DELL-G3:~$ more /proc/stat
cpu 24196 0 16846 1418083 0 541 0 0 0 0
cpu0 3837 0 3718 174835 0 432 0 0 0 0
cpu1 1157 0 1329 179904 0 10 0 0 0 0
cpu2 5576 0 3492 173323 0 21 0 0 0 0
cpu3 1815 0 1315 179260 0 14 0 0 0 0
cpu4 4073 0 2754 175564 0 29 0 0 0 0
cpu5 1978 0 1003 179410 0 9 0 0 0 0
cpu6 2701 0 1656 178034 0 12 0 0 0 0
cpu7 3059 0 1579 177753 0 14 0 0 0 0
intr 894498 377553 332 0 0 0 0 0 0 0 287178 0 0 1078 0 8572 189
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ctxt 494453
btime 1610953129
processes 725
procs_running 1
procs_blocked 0
softirq 531871 0 166655 5588 285920 8660 0 272 0 791 63985
```

Q2.

a.) Using the top command, the 1st line shows us that it is constant

```
top - 13:04:51 up 36 min, 0 users, load average: 0.52, 0.58, 0.59
Tasks: 11 total, 2 running, 5 sleeping, 4 stopped, 0 zombie
%Cpu(s): 22.0 us, 3.5 sy, 0.0 ni, 74.3 id, 0.0 wa, 0.1 hi, 0.0 si, 0.0 st
MiB Mem : 16244.6 total, 9757.8 free, 6262.7 used, 224.0 buff/cache
MiB Swap: 13336.2 total, 13204.4 free, 131.8 used. 9851.2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
762	sher	20	0	10404	444	360	R	99.0	0.0	1:12.30	cpu

PID = 762 [Infinite Loop]

b.) Using the above statistics, we can say that CPU consumes 99.0% and Memory consumption is 0.0% in this infinite loop.

c.) In the above output it can be seen:

Tasks: 11 total, 2 running, 5 sleeping, 4 stopped, 0 zombie

Current state of process can be seen in S column, it is Running.

Q3.

- a.) After executing cpu-print executable, it ran an infinite loop. To get the PID of this process, the command 'ps aux' was used. The output was as follows:

```
sher@DELL-G3:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   8936   312 ?        Ss   13:19    0:00 /init
root         8  0.0  0.0   8940   224 tty1     Ss   13:19    0:00 /init
sher        9  0.0  0.0  18096  3604 tty1     S    13:19    0:00 -bash
root       30  0.0  0.0   8940   224 tty2     Ss   13:26    0:00 /init
sher       31  0.0  0.0  18228  3832 tty2     S    13:26    0:00 -bash
sher       61 29.8  0.0  10536   580 tty2     R    13:27    0:17 ./cpu-print
sher       64  0.0  0.0  18648  1892 tty1     R    13:28    0:00 ps aux
sher@DELL-G3:~$
```

Thus, we can see its pid to be '61' for COMMAND './cpu-print'.

- b.) Here, using command 'ps -o ppid= -p 61', we can find parent of process 61, i.e, 31 and further its parent to be 30 that is init process.

```
sher@DELL-G3:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   8936   312 ?        Ss   13:19    0:00 /init
root         8  0.0  0.0   8940   224 tty1     Ss   13:19    0:00 /init
sher        9  0.0  0.0  18096  3604 tty1     S    13:19    0:00 -bash
root       30  0.0  0.0   8940   224 tty2     Ss   13:26    0:00 /init
sher       31  0.0  0.0  18228  3832 tty2     S    13:26    0:00 -bash
sher       61 29.9  0.0  10536   580 tty2     R    13:27    0:29 ./cpu-print
sher       65  0.0  0.0  18648  1892 tty1     R    13:29    0:00 ps aux
sher@DELL-G3:~$ ps -o ppid= -p 61
31
sher@DELL-G3:~$ ps -o ppid= -p 31
30
sher@DELL-G3:~$
```

- c.) After running command: './cpu-print > /tmp/tmp.txt &', a new process created with pid 71.

```
sher@DELL-G3:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   8936   312 ?        Ss   14:45    0:00 /init
root         8  0.0  0.0   8940   224 tty1     Ss   14:45    0:00 /init
sher        9  0.0  0.0  18096  3612 tty1     S    14:45    0:00 -bash
root       39  0.0  0.0   8940   224 tty2     Ss   14:56    0:00 /init
sher       40  0.0  0.0  18228  3836 tty2     S    14:56    0:00 -bash
sher       71 100  0.0  10536   572 tty2     R    14:58    0:12 ./cpu-print
sher       72  0.0  0.0  18648  1888 tty1     R    14:58    0:00 ps aux
```

Now, using command 'lsf -p 71' → we get following output,

```
sher@DELL-G3:~$ lsf -p 71
COMMAND  PID USER  FD  TYPE DEVICE SIZE      NODE NAME
cpu-print 71 sher  cwd  DIR    0,15  4096  7318349394478780 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code
cpu-print 71 sher  rtd  DIR    0,2  4096  5910874511227182 /
cpu-print 71 sher  txt  REG   0,15 16752 3096224743871685 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code/cpu-print
cpu-print 71 sher  mem  REG   0,0  480136 /usr/lib/x86_64-linux-gnu/libc-2.31.so (path dev=0,2, inode=2251799814165384)
cpu-print 71 sher  mem  REG   0,0  480120 /usr/lib/x86_64-linux-gnu/libd-2.31.so (path dev=0,2, inode=4222124651139960)
cpu-print 71 sher  mem  REG   0,0  54469 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code/cpu-print (path dev=0,15, inode=3096224743871685)
cpu-print 71 sher  0u  CHR  4,2  4785074604086670 /dev/tty2
cpu-print 71 sher  2u  CHR  4,2  4785074604086670 /dev/tty2
cpu-print 71 sher  1w  REG  0,2 3000971264 14073748835984248 /tmp/tmp.txt
```

Here, we can see that file descriptors are pointing to:

0(std input) → (Type is Character Special File) → Pointing to /dev/tty2  
2(std error) → (Type is Character Special File) → Pointing to /dev/tty2  
1(std output) → (Type is Regular File) → Pointing to /tmp/tmp.txt file

So, using > symbol in shell, it redirects the output to an output file in directory that can be seen using 'cat /tmp/tmp.txt'.



- d.) Running command './cpu-print | grep hello &' and then generating a process with pid 135 (Pipe process).

```

sher@DELL-G3:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  8936    312 ?        Ss   14:45   0:00 /init
root        11  0.0  0.0   8940    224 ?        Ss   14:45   0:00 /init
sher         9  0.0  0.0  18096   3612 tty1    S    14:45   0:00 -bash
root       110  0.0  0.0   1840    224 tty2    S    15:13   0:00 /init
sher       111  0.2  0.0   1828   3836 tty2    S    15:13   0:00 -bash
sher       134 96.2  0.0  10536    568 tty2    R    15:13   0:03 ./cpu-print
sher       135  6.5  0.0   1620   1288 tty2    S    15:13   0:00 grep --color=auto hello
sher       136  0.0  0.0   18648   1888 tty1    R    15:13   0:00 ps aux

sher@DELL-G3:~$ ls -l -p 135
COMMAND PID USER  FD  TYPE DEVICE SIZE      NODE NAME
grep    135 sher  cwd  DIR  0,15 4096 7318349394478780 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code
grep    135 sher  rtd  DIR  0,2 4096 5910974511227182 /
grep    135 sher  txt  REG  0,2 199136 1970324837282795 /usr/bin/grep
grep    135 sher  mem  REG  0,0 313283 /usr/lib/locale/C.UTF-8/LC_CTYPE (path dev=0,2, inode=3377699720841155)
grep    135 sher  mem  REG  0,0 313282 /usr/lib/locale/C.UTF-8/LC_COLLATE (path dev=0,2, inode=3096224744130498)
grep    135 sher  mem  REG  0,0 313294 /usr/lib/locale/locale-archive (path dev=0,2, inode=3096224744130510)
grep    135 sher  mem  REG  0,0 480149 /usr/lib/x86_64-linux-gnu/libpthread-2.31.so (path dev=0,2, inode=2251799814165397)
grep    135 sher  mem  REG  0,0 480136 /usr/lib/x86_64-linux-gnu/libc-2.31.so (path dev=0,2, inode=2251799814165384)
grep    135 sher  mem  REG  0,0 480137 /usr/lib/x86_64-linux-gnu/libdl-2.31.so (path dev=0,2, inode=3940649674429321)
grep    135 sher  mem  REG  0,0 328375 /usr/lib/x86_64-linux-gnu/libcres.so.3.13.3 (path dev=0,2, inode=1688849860592311)
grep    135 sher  mem  REG  0,0 313290 /usr/lib/locale/C.UTF-8/LC_NUMERIC (path dev=0,2, inode=1688849860577226)
grep    135 sher  mem  REG  0,0 313293 /usr/lib/locale/C.UTF-8/LC_TIME (path dev=0,2, inode=1688849860577229)
grep    135 sher  mem  REG  0,0 313288 /usr/lib/locale/C.UTF-8/LC_MONETARY (path dev=0,2, inode=1688849860577224)
grep    135 sher  mem  REG  0,0 313287 /usr/lib/locale/C.UTF-8/LC_MESSAGES/SYS_LC_MESSAGES (path dev=0,2, inode=1688849860577223)
grep    135 sher  mem  REG  0,0 313291 /usr/lib/locale/C.UTF-8/LC_PAPER (path dev=0,2, inode=1688849860577227)
grep    135 sher  mem  REG  0,0 480120 /usr/lib/x86_64-linux-gnu/lib-2.31.so (path dev=0,2, inode=422124651139960)
grep    135 sher  mem  REG  0,0 313289 /usr/lib/locale/C.UTF-8/LC_NAME (path dev=0,2, inode=1688849860577225)
grep    135 sher  mem  REG  0,0 313281 /usr/lib/locale/C.UTF-8/LC_ADDRESS (path dev=0,2, inode=5629499534526401)
grep    135 sher  mem  REG  0,0 313292 /usr/lib/locale/C.UTF-8/LC_TELEPHONE (path dev=0,2, inode=1688849860577228)
grep    135 sher  mem  REG  0,0 313285 /usr/lib/locale/C.UTF-8/LC_MEASUREMENT (path dev=0,2, inode=3377699720841157)
grep    135 sher  mem  REG  0,0 480550 /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache (path dev=0,2, inode=2251799814165798)
grep    135 sher  mem  REG  0,0 313284 /usr/lib/locale/C.UTF-8/LC_IDENTIFICATION (path dev=0,2, inode=422124651139960)
grep    135 sher  mem  REG  0,0 308203 /usr/bin/grep (path dev=0,2, inode=1970324837282795)
grep    135 sher  1u  CHR  4,2 4785074604086670 /dev/tty2
grep    135 sher  2u  CHR  4,2 4785074604086670 /dev/tty2
grep    135 sher  0r  FIFO 0,0 209 pipe

sher@DELL-G3:~$ ls -l -p 134
COMMAND PID USER  FD  TYPE DEVICE SIZE      NODE NAME
cpu-print 134 sher  cwd  DIR  0,15 4096 7318349394478780 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code
cpu-print 134 sher  rtd  DIR  0,2 4096 5910974511227182 /
cpu-print 134 sher  txt  REG  0,15 16752 3096224743871685 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code/cpu-print
cpu-print 134 sher  mem  REG  0,0 480136 /usr/lib/x86_64-linux-gnu/libc-2.31.so (path dev=0,2, inode=2251799814165384)
cpu-print 134 sher  mem  REG  0,0 480120 /usr/lib/x86_64-linux-gnu/lib-2.31.so (path dev=0,2, inode=422124651139960)
cpu-print 134 sher  mem  REG  0,0 54469 /mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 1/intro-code/cpu-print (path dev=0,15, inode=3096224743871685)
cpu-print 134 sher  0u  CHR  4,2 4785074604086670 /dev/tty2
cpu-print 134 sher  2u  CHR  4,2 4785074604086670 /dev/tty2
cpu-print 134 sher  1w  FIFO 0,0 209 pipe

sher@DELL-G3:~$

```

As seen in terminal output, that pid 135 is of pipe process and 134 is of './cpu-print' process.

PID 134 process writes to pipe with file descriptor 1 in FIFO Type and PID 135 process takes input from 134 pid process which doesn't write output as output not mapped in above command.

- e.) We use whereis command →

```

sher@DELL-G3:~/$ whereis {ls,cd,history,ps}
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
cd:
history: /usr/share/man/man3/history.3readline.gz
ps: /usr/bin/ps /usr/share/man/man1/ps.1.gz
sher@DELL-G3:~/$

```

It is clear that cd and history have no executable. history only has a man page. Thus, Built-in executables → cd, history  
Implemented by the bash code → ps, ls

Q4.

memory1.c → VmSize (10536 kB) & VmRSS (592 kB)

memory2.c → VmSize (10536 kB) & VmRSS (2544 kB)

VmSize is same as same size of int array initialized. But VmRSS is different because in memory2.c, we actually access that array.

VmRSS is the measure of how much RAM the process is actually using. VmSize includes RSS, plus things like shared libraries and memory mapped files (which don't actually use RAM), as well as allocated, but unused, memory.

Thus, Ram used more in memory2.c as array was actually being accessed and updated in it.

```
sher@DELL-G3:/$ more proc/239/status
```

```
Name: memory1
State: S (sleeping)
Tgid: 239
Pid: 239
PPid: 147
TracerPid: 0
Uid: 1000 1000 1000 1000
Gid: 1000 1000 1000 1000
FDSize: 3
Groups:
VmPeak: 0 kB
VmSize: 10536 kB
VmLck: 0 kB
VmHWM: 0 kB
VmRSS: 592 kB
VmData: 0 kB
VmStk: 0 kB
VmExe: 4 kB
VmLib: 0 kB
VmPTE: 0 kB
Threads: 1
SigQ: 0/0
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000000
SigCgt: 0000000000000000
CapInh: 0000000000000000
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 0000001fffffffff
Cpus_allowed: ff
Cpus_allowed_list: 0-7
Mems_allowed: 1
Mems_allowed_list: 0
voluntary_ctxt_switches: 150
nonvoluntary_ctxt_switches: 545
```

```
sher@DELL-G3:/$ more proc/238/status
```

```
Name: memory2
State: S (sleeping)
Tgid: 238
Pid: 238
PPid: 161
TracerPid: 0
Uid: 1000 1000 1000 1000
Gid: 1000 1000 1000 1000
FDSize: 3
Groups:
VmPeak: 0 kB
VmSize: 10536 kB
VmLck: 0 kB
VmHWM: 0 kB
VmRSS: 2544 kB
VmData: 0 kB
VmStk: 0 kB
VmExe: 4 kB
VmLib: 0 kB
VmPTE: 0 kB
Threads: 1
SigQ: 0/0
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000000
SigCgt: 0000000000000000
CapInh: 0000000000000000
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 0000001fffffffff
Cpus_allowed: ff
Cpus_allowed_list: 0-7
Mems_allowed: 1
Mems_allowed_list: 0
voluntary_ctxt_switches: 150
nonvoluntary_ctxt_switches: 545
```

Q5.

Run command `iostat -xtc 1` for all stats regarding to disk.

Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	115.00	699.00	143.00	60008.00	1088.00	145.12	2.27	2.70	2.12	5.54	1.13	95.20
dm-0	0.00	0.00	0.00	272.00	0.00	1088.00	8.00	2.56	9.40	0.00	9.40	0.29	8.00
07/21/2017 11:20:54 PM													
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle							
	0.00	0.00	6.78	6.41	0.00	86.81							
Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	178.00	763.00	183.00	65152.00	1456.00	140.82	1.93	2.04	1.88	2.73	0.99	93.60
dm-0	0.00	0.00	0.00	364.00	0.00	1456.00	8.00	1.40	3.84	0.00	3.84	0.13	4.80
07/21/2017 11:20:55 PM													
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle							
	0.25	0.00	6.26	6.88	0.00	86.61							
Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	136.00	722.00	135.00	61936.00	1112.00	147.14	2.97	3.47	2.05	11.05	1.10	94.40
dm-0	0.00	0.00	0.00	278.00	0.00	1112.00	8.00	3.47	12.47	0.00	12.47	0.32	8.80

While './disk' is running the disk utilization is close to 95% .

Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
07/21/2017 11:23:12 PM													
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle							
	5.38	0.00	7.12	0.12	0.00	87.38							
Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
07/21/2017 11:23:13 PM													
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle							
	3.76	0.00	4.39	0.00	0.00	91.85							
Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

While running './disk1', the idleness of the cpu is not 100% so we know a process is running, but the disk utilization is ~0% (can't be seen on the 1 second scale). Once the file is in the cache, it need not read from the disk.

This difference is because './disk' is reading all the different files while './disk1' is reading only 1 file again and again. This 1 file is in the cache now so fast read. While 5000 files can't be put in cache so have to be read from the disk again and again.