

1)

a) processor: the index of the processor in the system
cores : the number of cores in the processor

b) 8

c) 4

d) Processor 0: 1213.4 MHz
Processor 1: 911.98 MHz
Processor 2: 897.98 MHz
Processor 3: 955.74 Mhz

e) 8086608 Kb

f) 3240124 Kb

g) 11652

h) 1058548351

2)

a) 12131

b) Ids (Starting from child and going up the hierarchy):

12131

12115

12107

10175

1

c) Standard Input (0) --> /dev/pts/2

Standard Output(1) --> /tmp/tmp.txt

Standard Error(2) --> /dev/pts/2

The standard input for this process is the terminal (/dev/pts2) and the output for this process is directed to the *tmp.txt* file in tmp directory. Thus, by changing the file descriptors, the I/O redirection is handled by shell

d) Newly Spawned Process – 12434 and 12435

For process 12434:-

0 ---> /dev/pts/2

1 ---> 'pipe:[158010]'

2 ---> /dev/pts/2

For process 12435:-

0 ---> 'pipe:[158010]'

1 ---> /dev/pts/2

2 ---> /dev/pts/2

The output of first process points to pipe having id 158010 which in turn acts as input for the second process.

e) ls and ps are executed by the bash shell
cd and history are implemented by the bash shell
as seen by which command

3) memory1.c ---> Virtual:8296 Resident:756
memory2.c ---> Virtual:8292 Resident:3212

In both memory 1 and memory 2 the arrays have been initialized and hence their virtual memory is same. But in memory2 some members of the array have been allocated a value and hence the resident memory is more for memory2

4) When disk process is run and parallelly iostat 5 is also run, we observe that the kb_read keeps on increasing since the program reads pdf files randomly and cache is not enough to store the files.
For disk: Memory utilization : 91.2%

When disk1 process is run and parallelly iostat 5 is also run, we observe that the kb_read becomes constant after some time since the program keeps on reading the same file so the file is stored in the cache and is not read again.
For disk1: Memory Utilization: 1.8%

5) For cpu: CPU is the bottleneck since the percentage CPU utilization is 100%. As seen in the program, it is an infinite loop that stores the value from 1 to 10! in a variable thus memory is not a problem but due to infinite utilization, the CPU usage is high.

For disk: Memory is the bottleneck (as seen by iostat command) since the program reads the files again and again and the disk is being used continuously in which read/write operation is slow.

For disk1: CPU is the bottleneck since the program reads only one file which can be stored in the cache so memory is not an issue and since it is an infinite loop CPU utilization is high(100%)