

# Operating System

## Worksheet 1

### Introduction to Linux Tools

Bhavya Mithal

2110110165

bm820

1.

- a) The command "more /proc/cpuinfo" displays information about the processors on your machine.

The term "processor" refers to a physical processing unit, which is the chip that contains the CPU.

A core refers to a separate processing unit within a physical processor chip, typically each core can execute its own set of instructions simultaneously.

The command "lscpu" can also be used to verify the information about the processors and cores on your machine.

CPU hyper-threading is a technology that allows a single physical processor to appear as multiple logical processors, which can improve performance by allowing multiple threads to run on separate logical processors.

The screenshot shows the Visual Studio Code interface with a terminal window. The left sidebar shows a file explorer with a directory structure including 'OS', 'OS - files', 'Lecture Slides', and various C files. The terminal window has two tabs: 'TERMINAL' and 'OUTPUT'. The 'TERMINAL' tab is active, showing the output of the command 'more /proc/cpuinfo'. The output lists various CPU parameters for an Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, including processor, vendor\_id, cpu\_family, model, model\_name, stepping, microcode, cpu MHz, cache size, physical\_id, siblings, core\_id, cpu cores, apicid, initial apicid, fpu, fpu\_exception, cpuid\_level, wp, and flags. The 'OUTPUT' tab shows the output of the command 'lscpu', which provides a summary of the system's CPU architecture, including Architecture (x86\_64), CPU op-mode(s) (32-bit, 64-bit), Byte Order (Little Endian), Address sizes (36 bits physical, 48 bits virtual), CPU(s) (8), On-line CPU(s) list (0-7), Thread(s) per core (2), Core(s) per socket (4), Socket(s) (1), Vendor ID (GenuineIntel), CPU family (6), Model (142), Model name (Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz), Stepping (11), CPU MHz (1800.000), CPU max MHz (1800.0000), Bogomips (2600.00), Virtualization (VT-x), Hypervisor vendor (Windows Subsystem for Linux), and Virtualization type (container). The flags are also listed.

- b) To find out how many cores your machine has, you can use the command "lscpu" and look for the "Core(s) per socket" field in the output. **Here, Cores per socket = 4**
- c) To find out how many processors your machine has, you can use the command "lscpu" and look for the "Socket(s)" field in the output. **Here, Processors = 1**
- d) To find out the frequency of each processor, you can use the command "more /proc/cpuinfo" and look for the "cpu MHz" field in the output. **Here, Frequency = 1800 MHz.**
- e) To find out the architecture of your CPU, you can use the command "lscpu" and look for the "Architecture" field in the output. **Here, Architecture = x86\_64**
- f) To find out how much physical memory your system has, you can use the command "free -h" and look for the "Mem" field in the output. **Here, Memory = 7.9 GB**
- g) To find out how much of this memory is free, you can use the command "free -h" and look for the "Mem Available" or "Mem Free" fields in the output. **Here, Free memory = 3.6 GB**
- h) To find out the total number of forks and context switches since the system booted up, you can use the command "vmstat" and look for the "forks" and "cs" fields in the output.

2. a) PID is 349
  - b) CPU – 100%
  - Memory – 0%
  - c) Running State

```

top - 22:16:19 up 1:29, 0 users, load averages: 0.52, 0.58, 0.
Tasks: 9 total, 2 running, 5 sleeping, 2 stopped, 0 sh
%Cpu(s): 17.8 us, 8.7 sy, 0.0 ni, 81.5 id, 0.0 wa, 0.0 hi,
Mem Mem : 8879.2 total, 3180.7 free, 4673.8 used, 224.0
Mem Swap: 15174.0 total, 15068.0 free, 106.0 used, 3274.0

  PID USER      PR  NI   VIRT    RES    SHR   S %CPU  %MEM
 349 bhavya    20   0 10404    444    328   R 100.3   0.0
  1 root       20   0   8952    168    116   S   0.0   0.0
 11 root       20   0   8952    128    92   S   0.0   0.0
 12 bhavya    20   0 10888    884    884   S   0.0   0.0
 25 root       20   0   8952    128    92   S   0.0   0.0
 26 bhavya    20   0 10888    896    816   S   0.0   0.0
 39 bhavya    20   0 15488    524    512   T   0.0   0.0
 358 bhavya    20   0 10968    288    152   T   0.0   0.0
 351 bhavya    20   0 10924    200    152   R   0.0   0.0

```

3. a) PID = 105
  - b) Parent PID = 91
 

Ancestor PID →

```

init(1) —┬─ init(11) —┬─ bash(12) —┬─ cpu-print(105)
          │             └─ init(91) —┬─ bash(92) —┬─ pstree(113)
          └─ {init}(10)

```

The screenshot shows a VS Code terminal window with the following content:

```
File Edit Selection View Go Run Terminal Help
OS - Visual Studio Code

TERMINAL  PROCESSING  OUTPUT  DEBUG CONSOLE

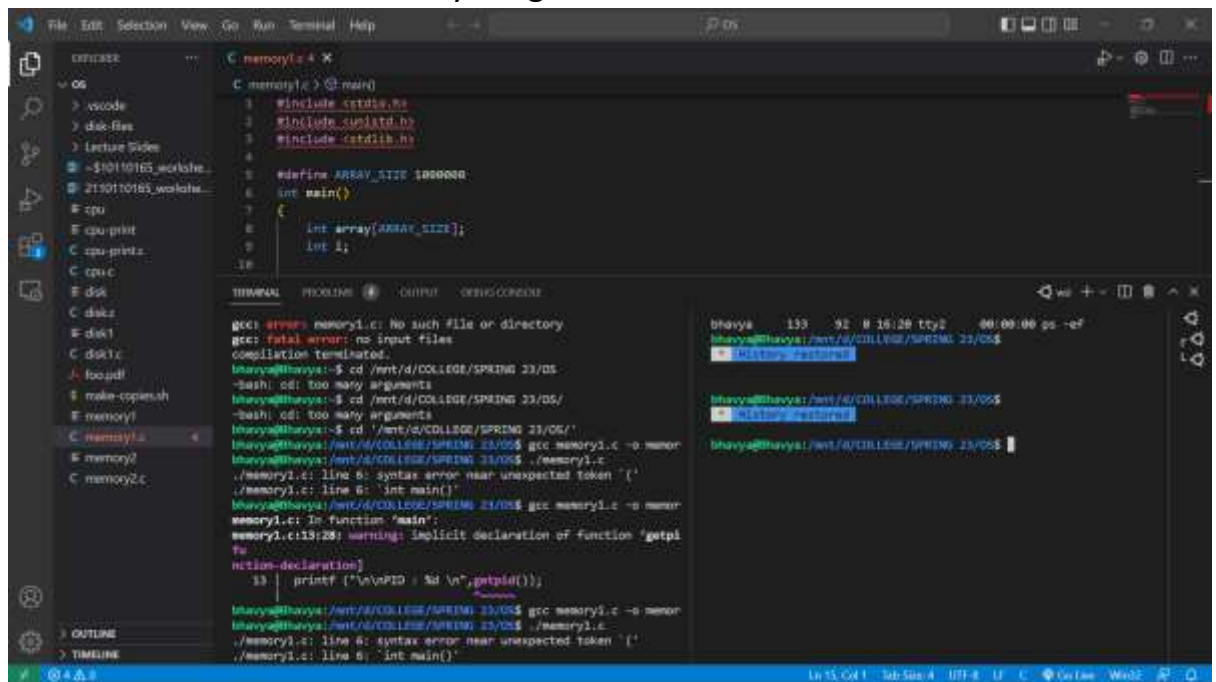
1674728639 sec, 481365 usec
1674728639 sec, 481544 usec
1674728639 sec, 481730 usec
1674728639 sec, 481913 usec
1674728639 sec, 482130 usec
1674728639 sec, 482353 usec
1674728639 sec, 482537 usec
1674728639 sec, 482720 usec
1674728639 sec, 482903 usec
1674728639 sec, 483087 usec
1674728639 sec, 483274 usec
1674728639 sec, 483467 usec
1674728639 sec, 483741 usec
1674728639 sec, 484015 usec
1674728639 sec, 484274 usec
1674728639 sec, 484498 usec
1674728639 sec, 484729 usec
1674728639 sec, 484948 usec
1674728639 sec, 485189 usec
1674728639 sec, 485400 usec
1674728639 sec, 485590 usec
1674728639 sec, 485707 usec
1674728639 sec, 485754 usec
1674728639 sec, 485785 usec
1674728639 sec, 485728 usec
1674728639 sec, 486131 usec
1674728639 sec, 486588 usec
1674728639 sec, 486868 usec
1674728639 sec, 486927 usec
1674728639 sec, 486741 usec
1674728639 sec, 486289 usec
1674728639 sec, 486445 usec
1674728639 sec, 486628 usec
1674728639 sec, 486823 usec
1674728639 sec, 486840 usec
1674728639 sec, 486156 usec
1674728639 sec, 486423 usec
1674728639 sec, 486614 usec
1674728639 sec, 486189 usec

shavya@shavya:/mnt/d/COLLEGE/SPRING_23/OS$ ps -e | grep cpu-print
105  tty1      00:01:45  cpu-print
shavya@shavya:/mnt/d/COLLEGE/SPRING_23/OS$ ps -o ppid
PPID
  51
  52
shavya@shavya:/mnt/d/COLLEGE/SPRING_23/OS$ pstree -p
init(1)---init(11)---bash(12)---cpu-print(105)
      |---init(9)---bash(92)---pstree(122)
      |---init(10)
shavya@shavya:/mnt/d/COLLEGE/SPRING_23/OS$
```

c) I/O redirection is a feature of the shell that allows a user to redirect the input or output of a command to a file or another source. This can be achieved by using the ">" or ">>" operators to redirect the standard output of a command to a file and the "<" operator to redirect the standard input of a command from a file. When a user runs a command with I/O redirection, the shell creates a new process to execute the command, and then modifies the file descriptor table of the process to redirect the input or output as specified. For example, when a user runs the command `./cpu-print > /tmp/tmp.txt &`, the shell creates a new process to execute the `cpu-print` program and then modifies the file descriptor table of the process so that file descriptor 1 (standard output) points to the file `/tmp/tmp.txt`. This causes all output generated by the program to be written to the file instead of the terminal. The `&` at the end of the command runs the command in background so that we can continue with other command in terminal.

e) The commands `"cd"`, `"history"` and `"ps"` are built-in commands that are implemented by the bash shell itself, while `"ls"` is an executable that already comes with the Linux kernel installation.

4. I was unable to do the 4<sup>th</sup> question, as when I ran the memory files, it showed that they both had some error and I tried to google to resolve the issue but couldn't find anything useful.



The screenshot shows a VS Code editor with a C program named `memory1.c` open. The program includes `stdio.h`, `unistd.h`, and `stdlib.h`, defines `ARRAY_SIZE` as 1000000, and has a `main` function that declares an array and a variable `i`. The terminal shows the following errors:

```
gcc: error: memory1.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
bhavya@bhavya:~$ cd /mnt/d/COLLEGE/SPRING 23/OS/
-bash: cd: too many arguments
bhavya@bhavya:~$ cd /mnt/d/COLLEGE/SPRING 23/OS/
-bash: cd: too many arguments
bhavya@bhavya:~$ cd /mnt/d/COLLEGE/SPRING 23/OS/
bhavya@bhavya:/mnt/d/COLLEGE/SPRING 23/OS$ gcc memory1.c -o memor
bhavya@bhavya:/mnt/d/COLLEGE/SPRING 23/OS$ ./memory1.c
./memory1.c: line 6: syntax error near unexpected token '('
./memory1.c: line 6: 'int main()'
bhavya@bhavya:/mnt/d/COLLEGE/SPRING 23/OS$ gcc memory1.c -o memor
bhavya@bhavya:/mnt/d/COLLEGE/SPRING 23/OS$ ./memory1.c
./memory1.c: line 6: syntax error near unexpected token '('
./memory1.c: line 6: 'int main()'
memory1.c:13:28: warning: implicit declaration of function 'getpid'
      fu
      ^
      action-declaration]
13 | printf("\n\nPID : %d\n",getpid());
    |
```

5. When we run the program `disk.c`, it reads all the files in the "disk-files" folder sequentially. Since all the files are identical and are read sequentially, the disk utilization would be consistent throughout the program's execution. We can use the command "iostat" to measure the disk utilization while the program is running and observe the read operations, read KB/s, and the %utilization. When we run the program `disk1.c`, it reads all the files in the "disk-files" folder randomly. Since the files are read randomly, the disk utilization would be less consistent throughout the program's execution. we can use the command "iostat" to measure the disk utilization while the program is running and observe the read operations, read KB/s, and the %utilization.

The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of the `lsblk` command, showing disk usage and system information. The output is as follows:

```
thavya@thavya:/mnt/d/COLLEGE/SPRING 23/01$ lsblk
Linux 4.4.0-19041-Microsoft (thavya) 01/27/23 _x86_64_ (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    4.38    0.00    0.00   95.61

Device            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
vda                  0             0              0              0            0            0
vda1                  0             0              0              0            0            0

thavya@thavya:/mnt/d/COLLEGE/SPRING 23/01$ lsblk
Linux 4.4.0-19041-Microsoft (thavya) 01/27/23 _x86_64_ (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    4.38    0.00    0.00   95.61

Device            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
vda                  0             0              0              0            0            0
vda1                  0             0              0              0            0            0

thavya@thavya:/mnt/d/COLLEGE/SPRING 23/01$
```