Operating System

Worksheet 1

Introduction to Linux Tools

Bhavya Mithal

2110110165

bm820

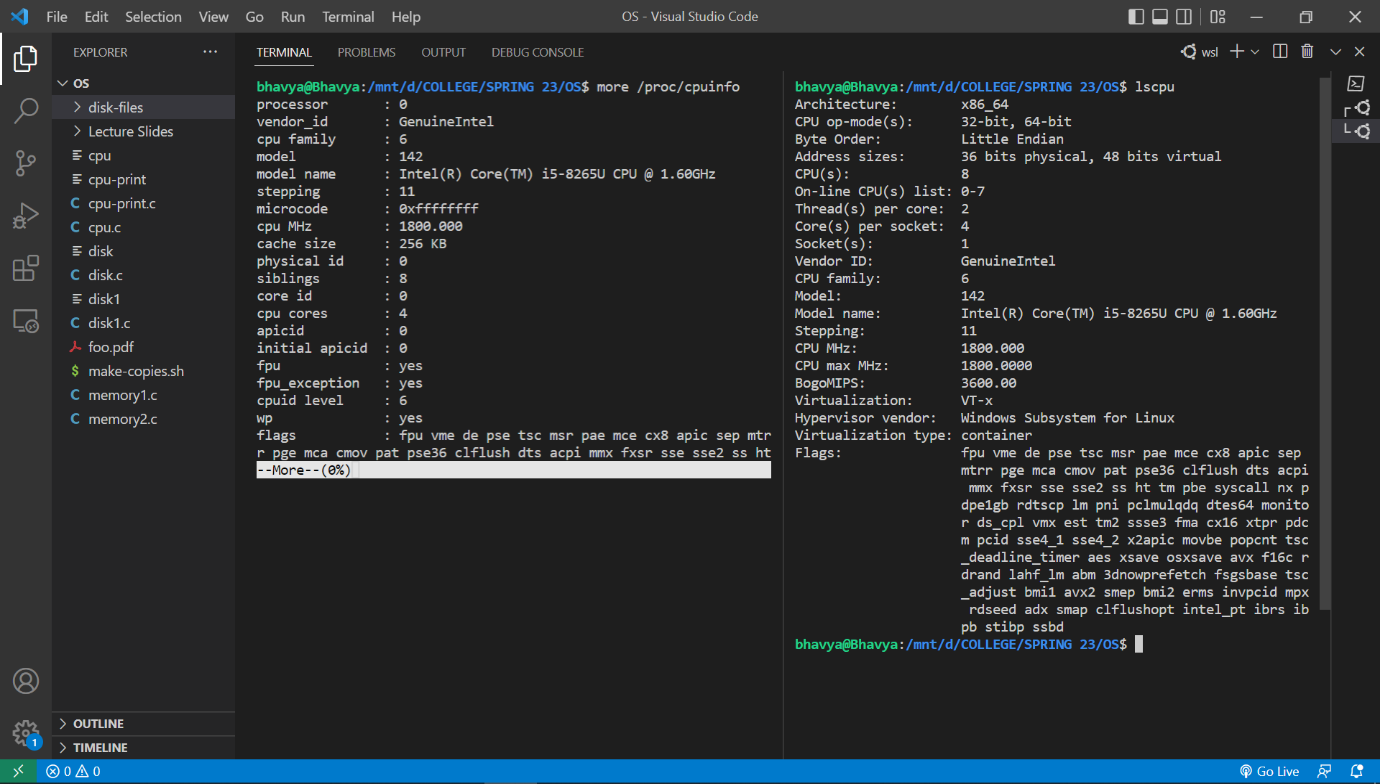
1. The command "more /proc/cpuinfo" displays information about the processors on your machine.

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.

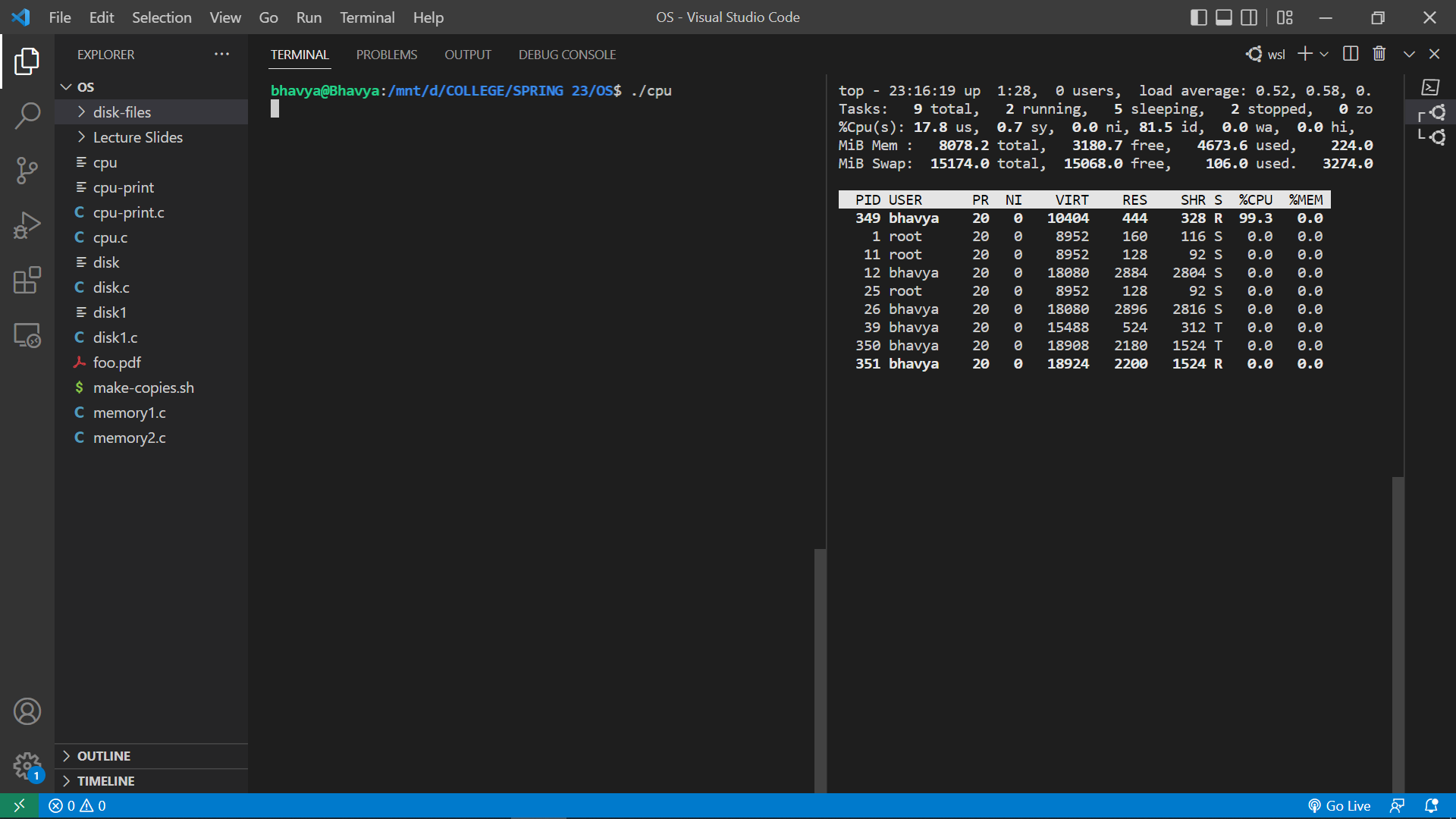


1. To find out how many cores your machine has, you can use the command "lscpu" and look for the "Core(s) per socket" field in the output**. Here, Cores per socket = 4**
2. To find out how many processors your machine has, you can use the command "lscpu" and look for the "Socket(s)" field in the output. **Here, Processors = 1**
3. 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.**
4. 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**
5. 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**
6. 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**
7. 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.
8. a) PID is 349

b) CPU – 100%

Memory – 0%

c) Running State



1. a) PID = 105

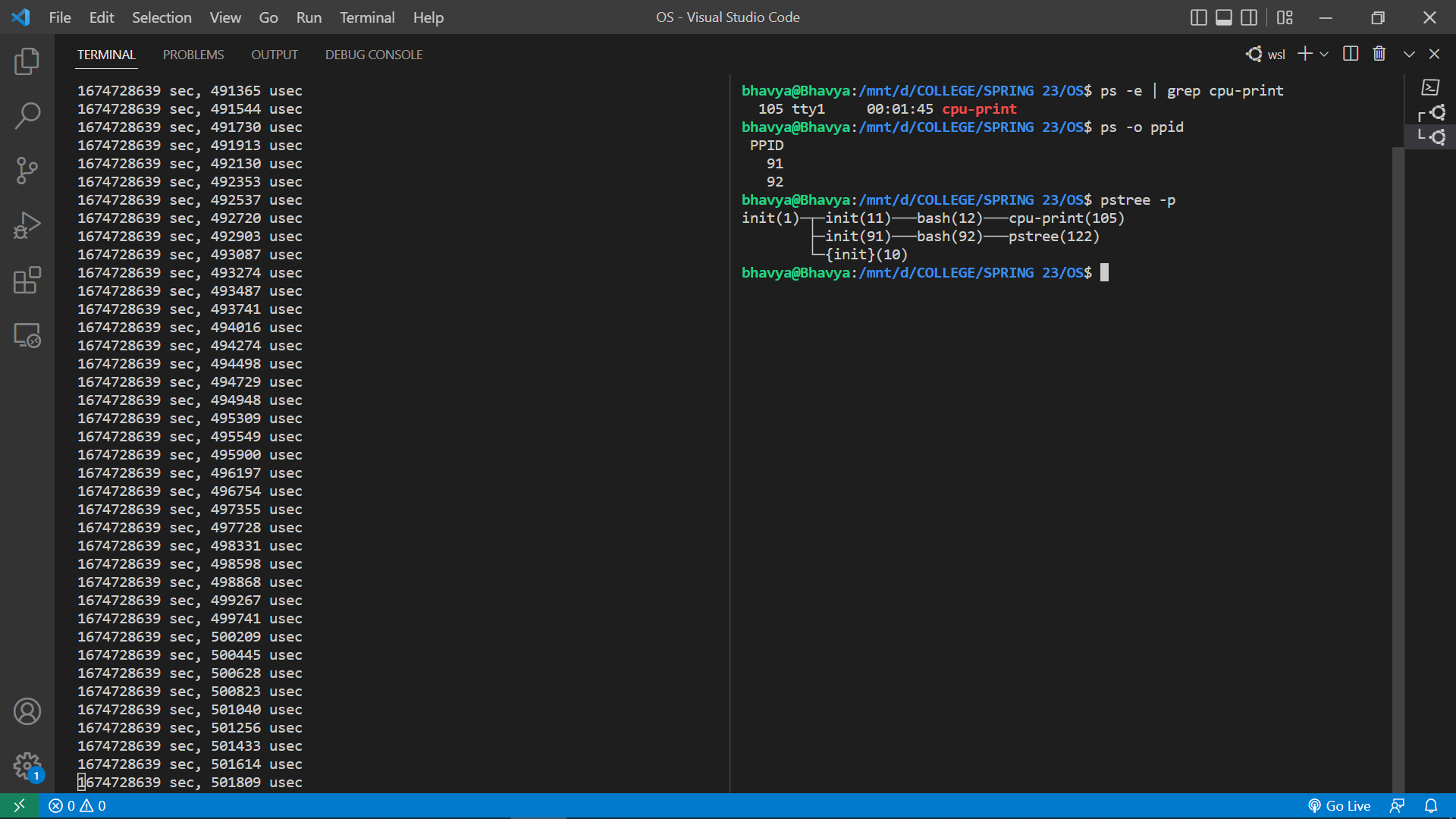
b) Parent PID = 91

Ancestor PID 🡪

init(1)─┬─init(11)───bash(12)───cpu-print(105)

├─init(91)───bash(92)───pstree(113)

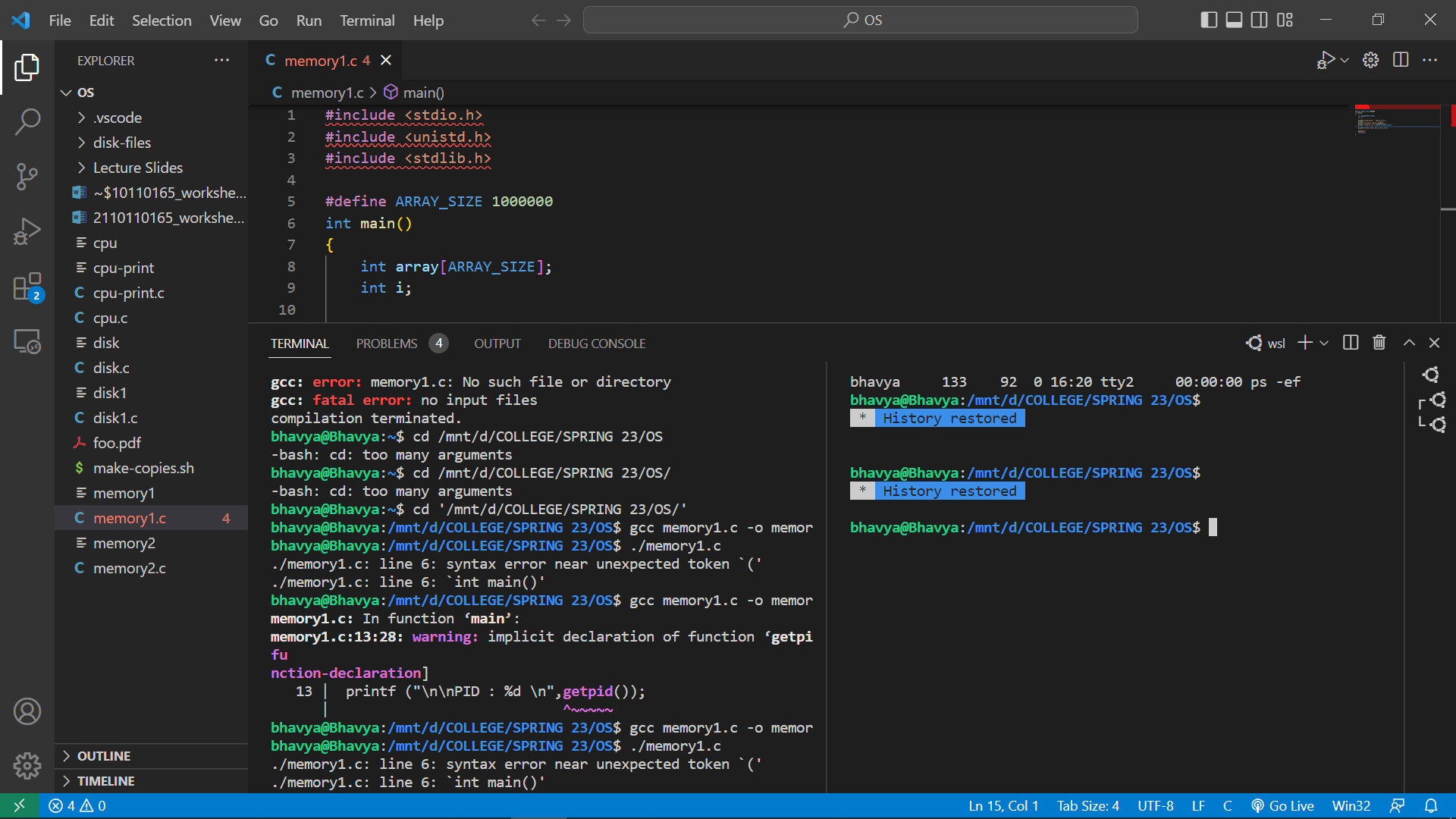
└─{init}(10)



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.

1. I was unable to do the 4th 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.



1. 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.

