****

**RAMANUJAN COLLEGE**

(University of Delhi)

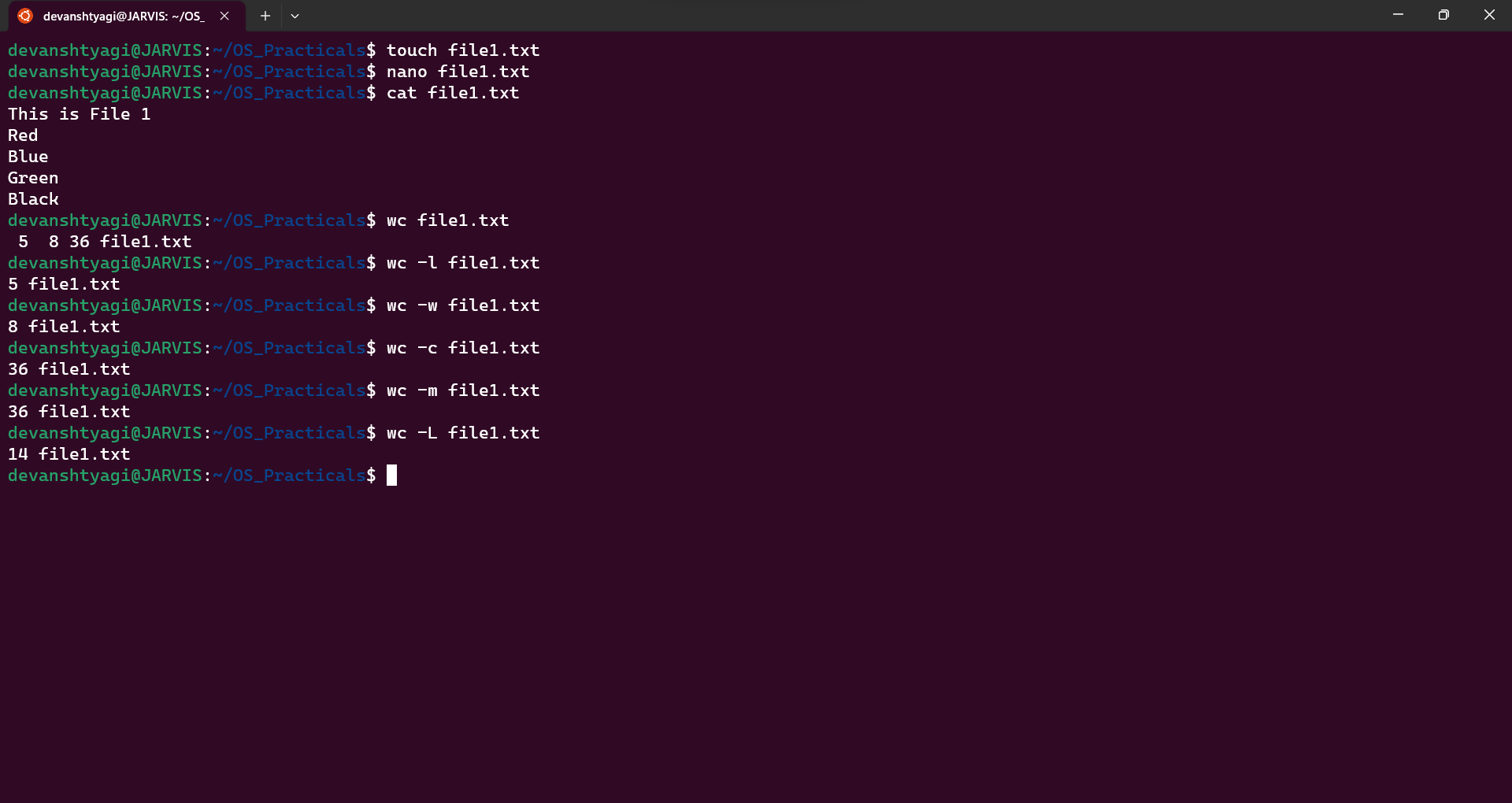
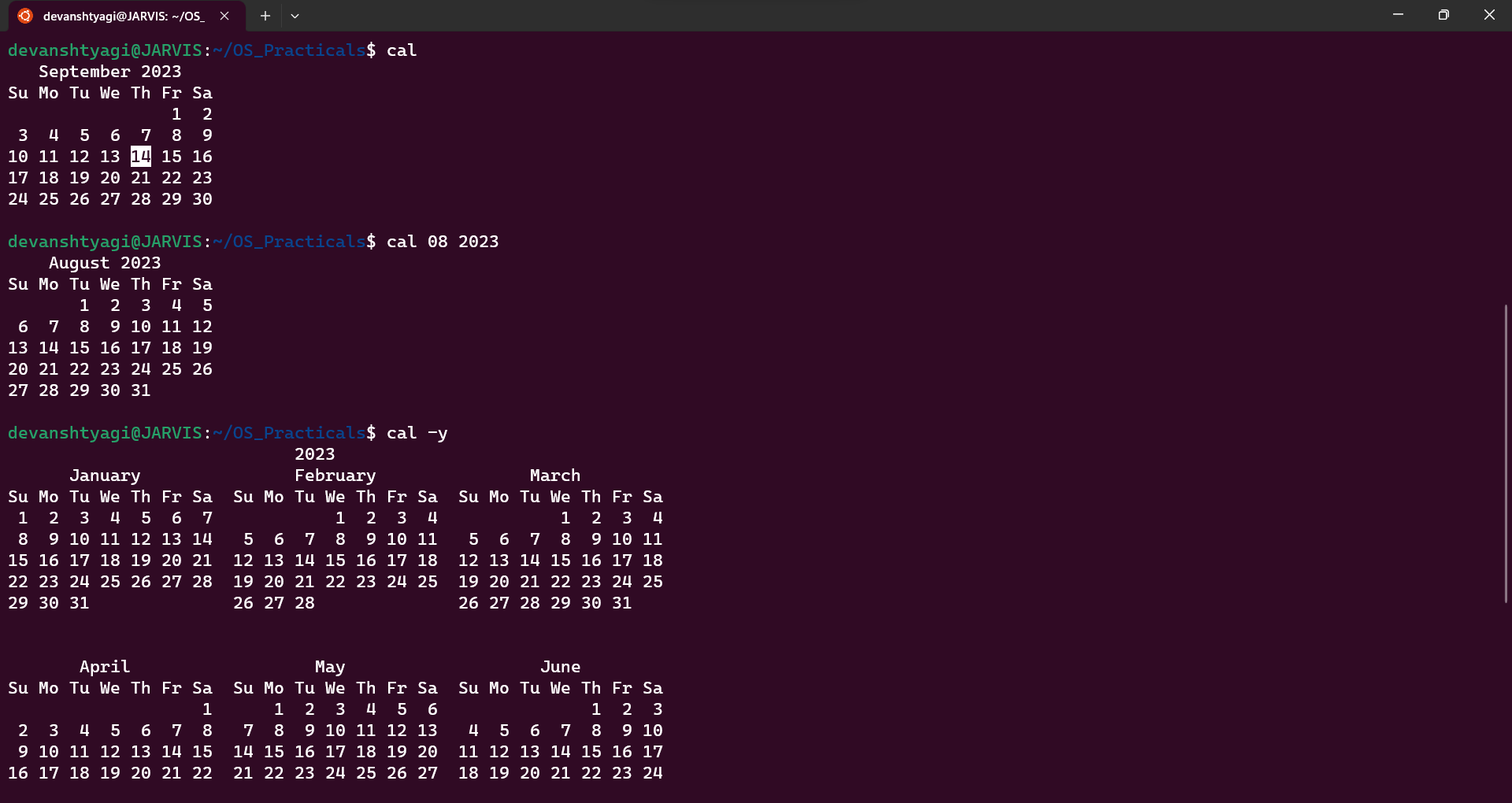
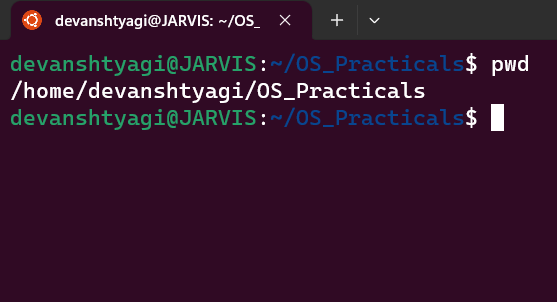
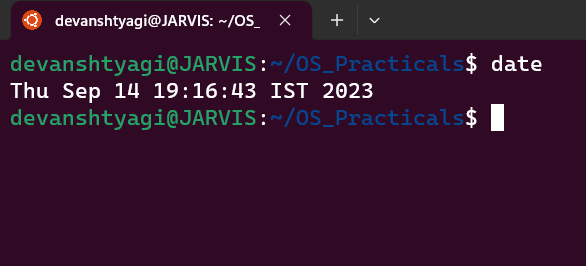
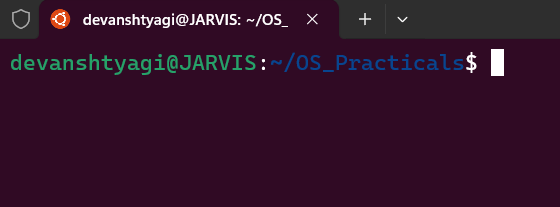
**OPERATING SYSTEM**

**(PRACTICAL FILE)**

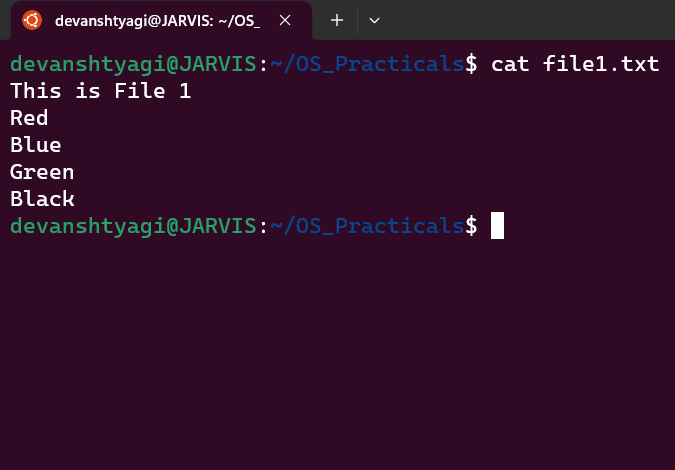
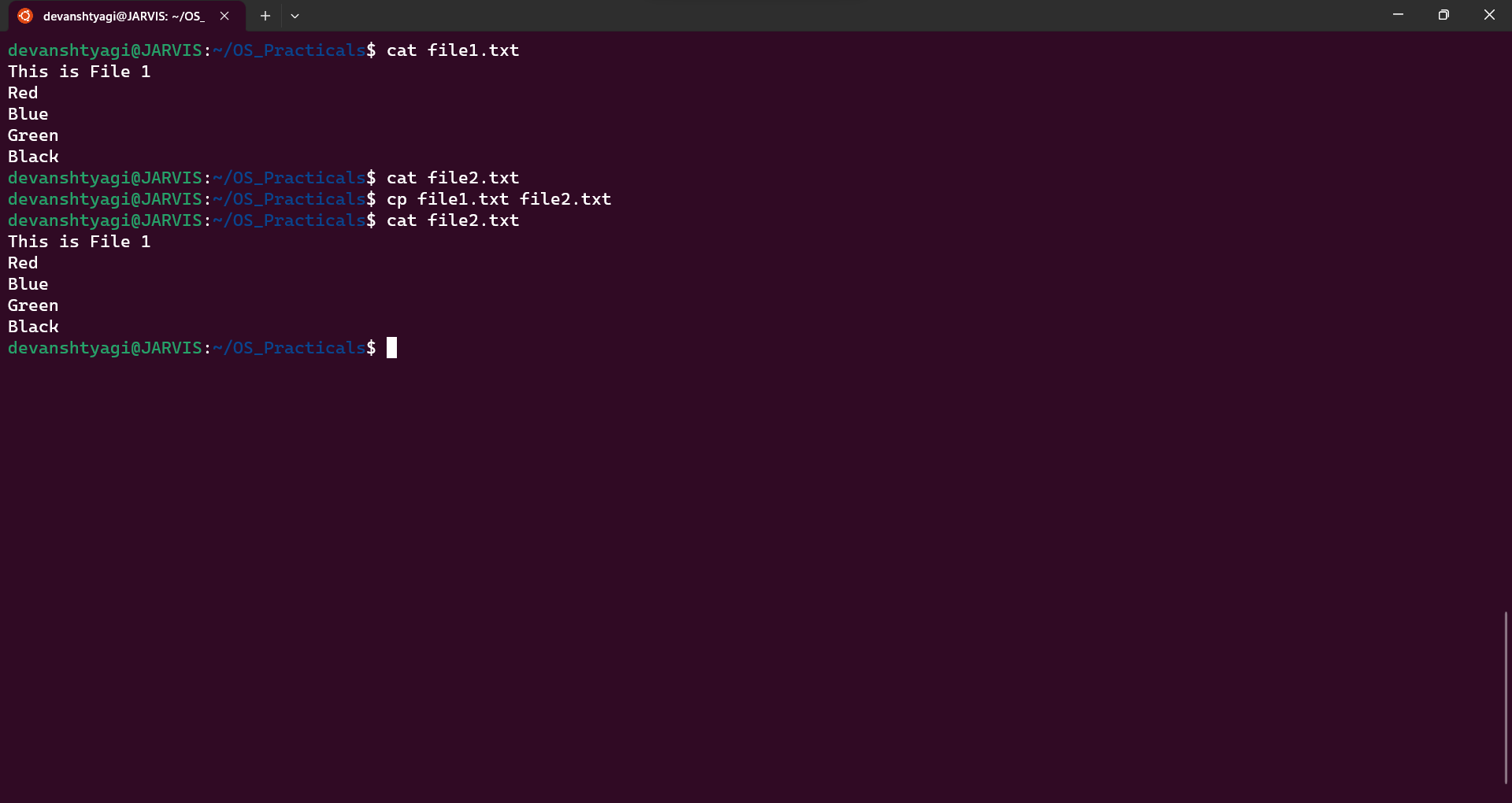
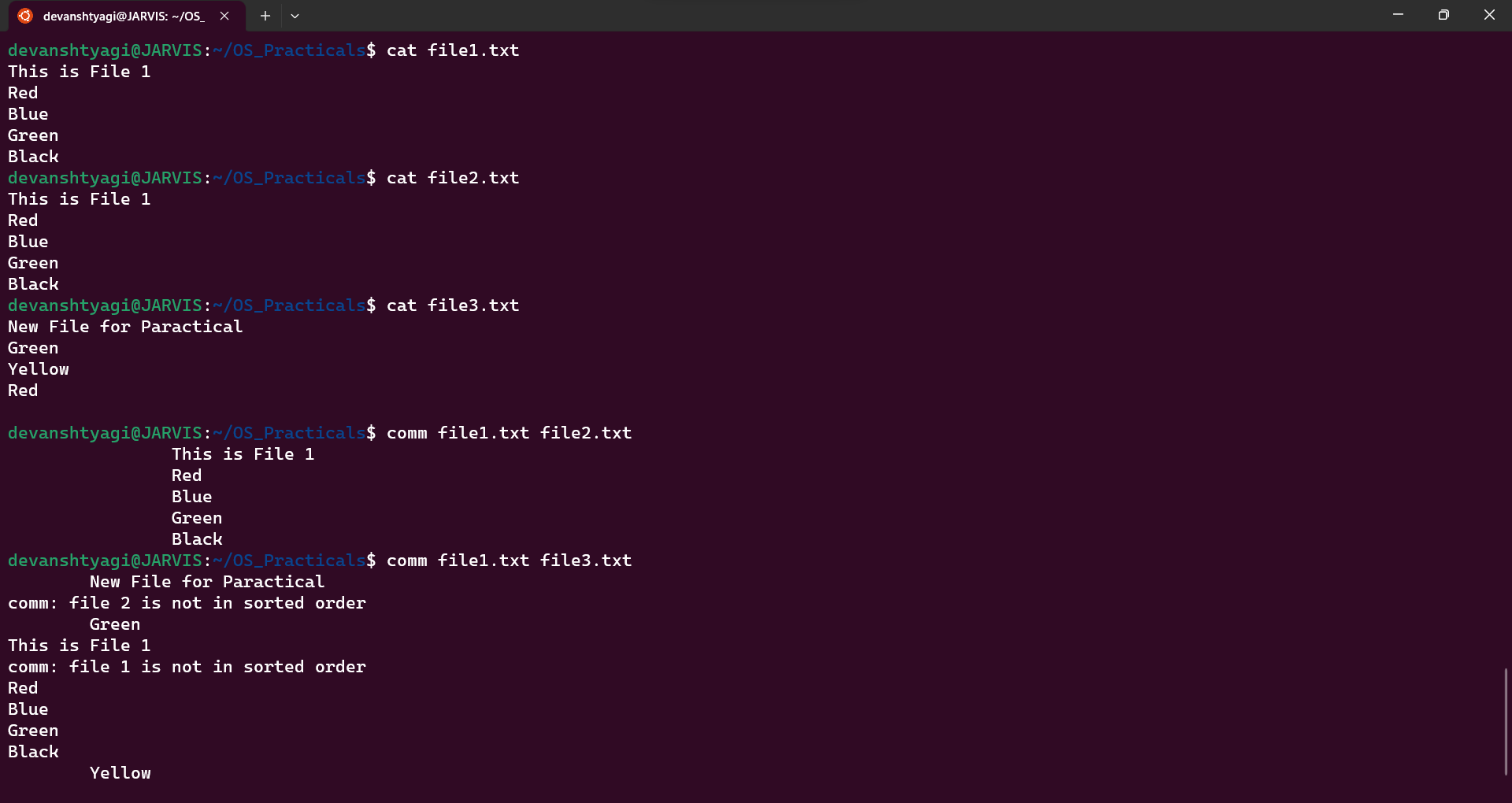
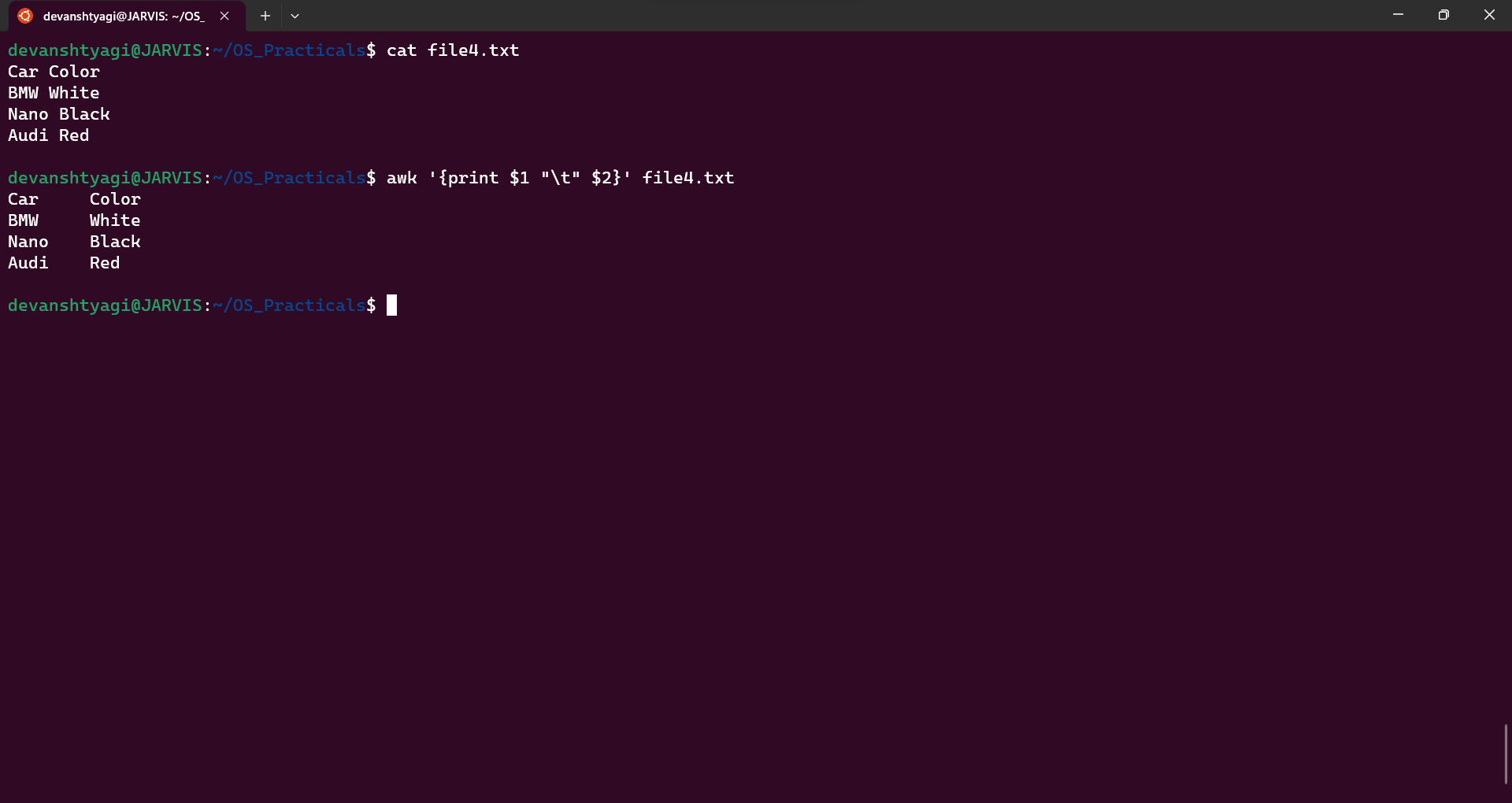
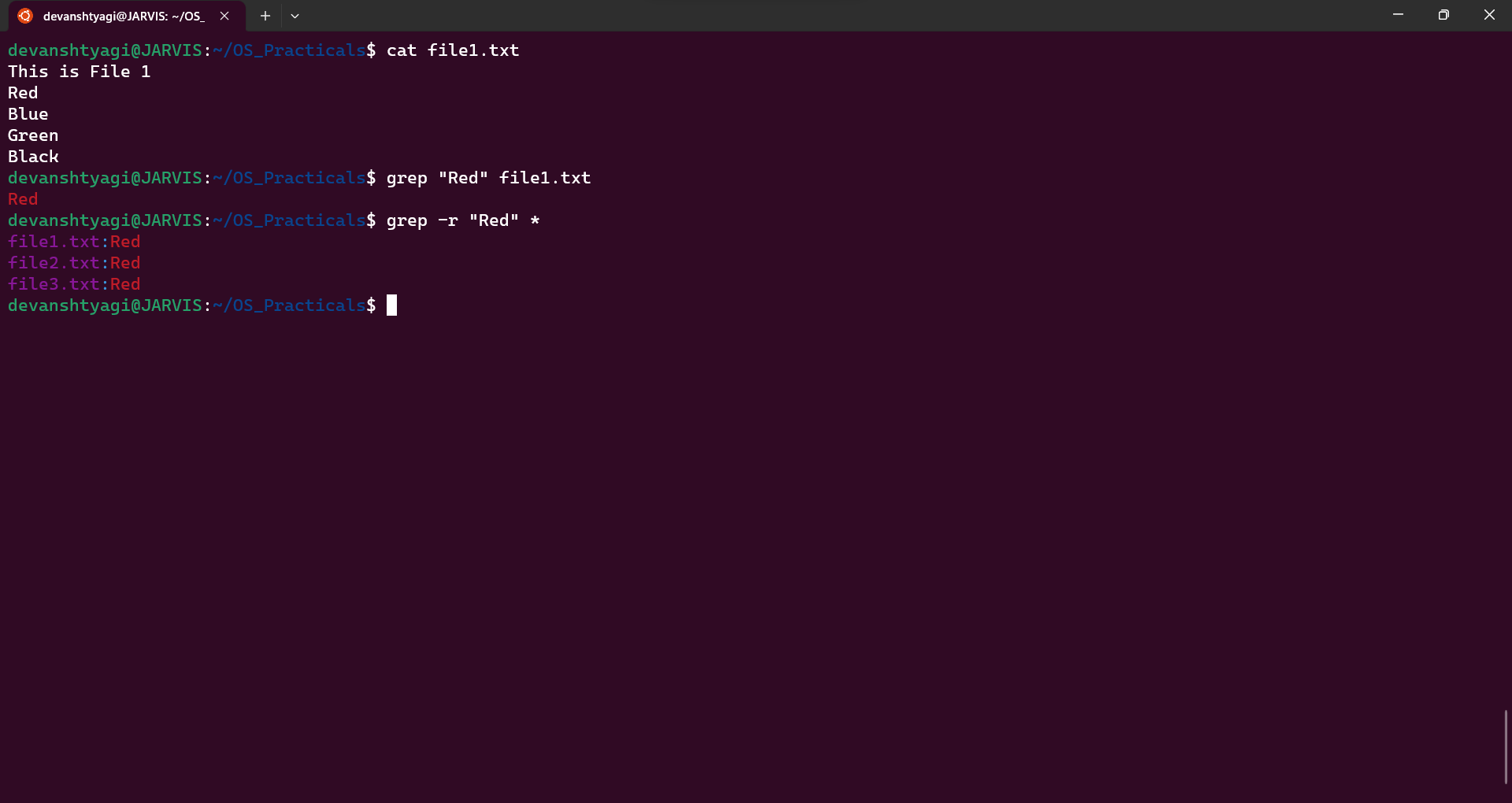
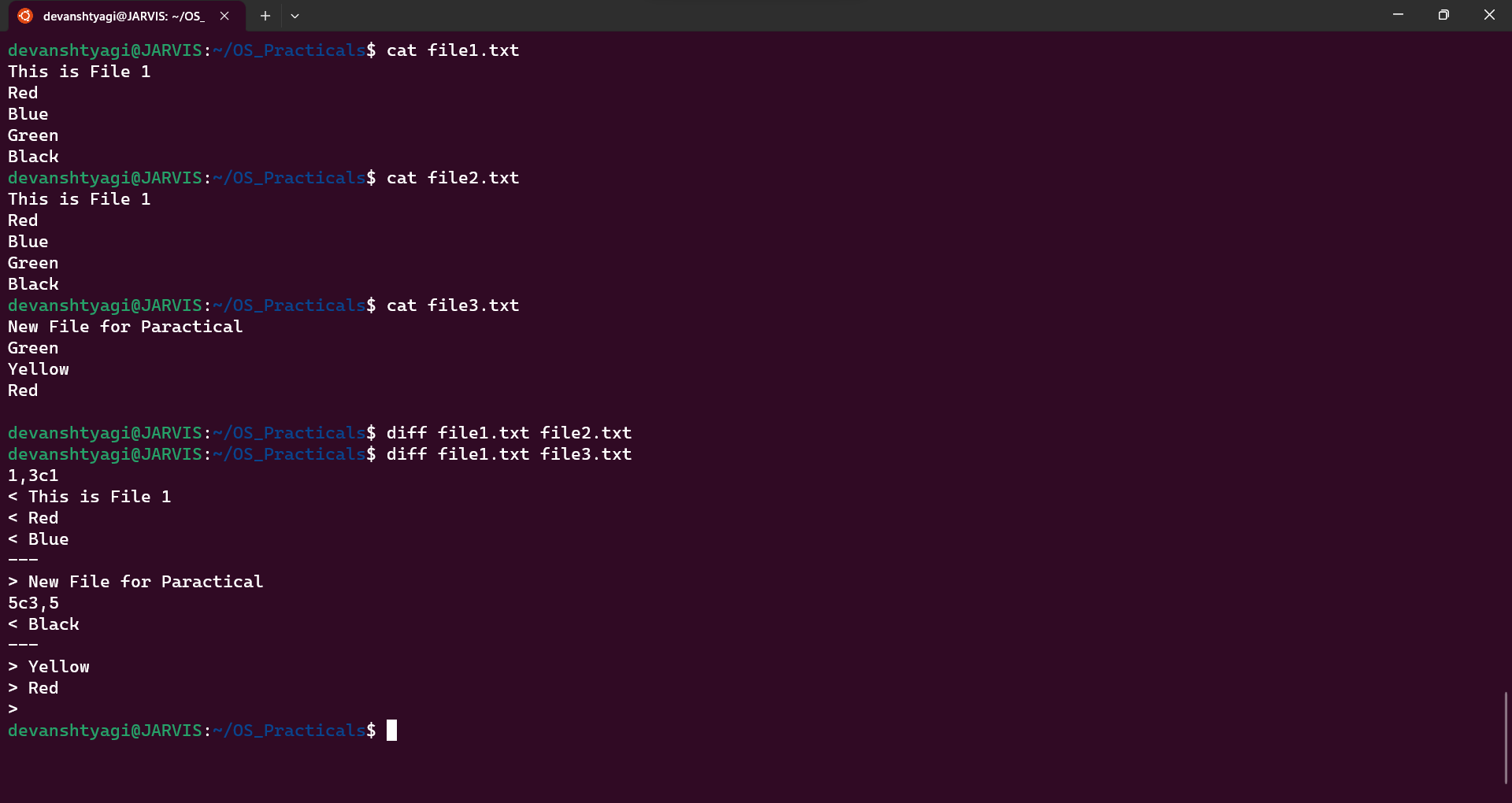
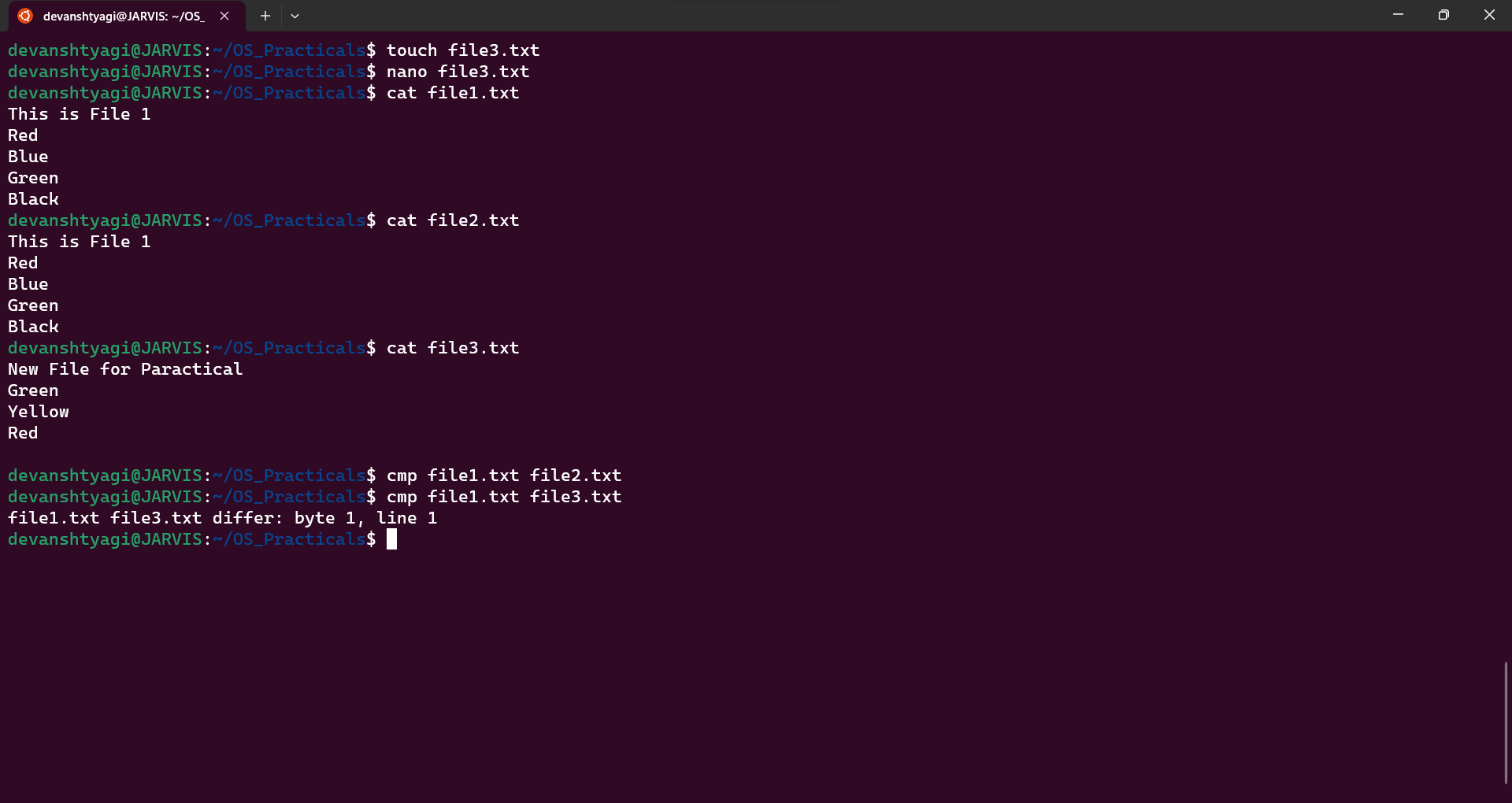
|  |  |  |
| --- | --- | --- |
| **Name :** | Devansh Tyagi |  |
| **College Roll No. :** | 20221416 |  |
| **University Roll No. :** | 22020570005 |  |
| **Course :** | B.Sc. (Hons.) Computer Science |  |
| **Year :** | Second |  |
|  |  |  |

1. Execute various LINUX commands for:

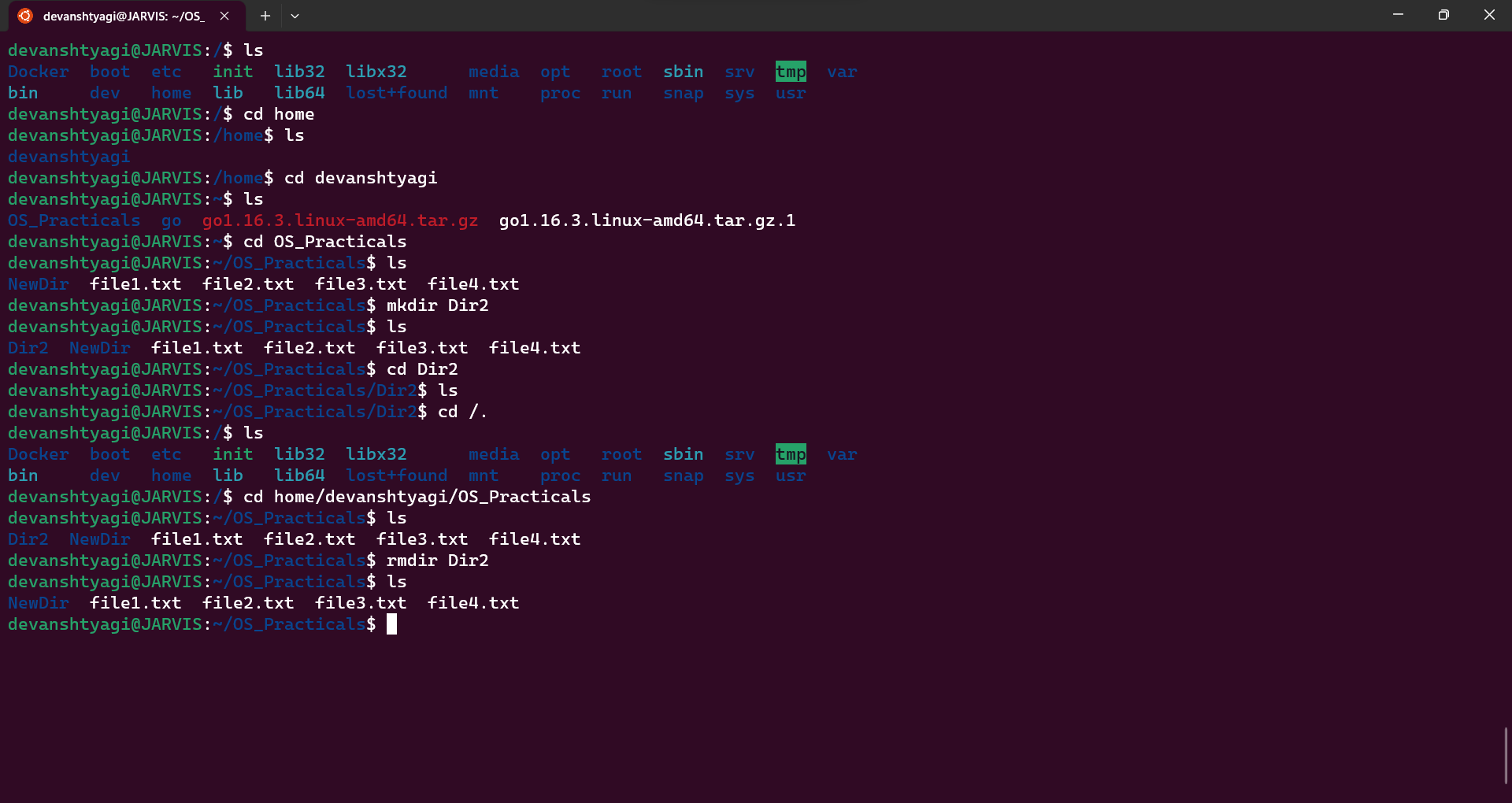
i. Information Maintenance: wc, clear, cal, who, date, pwd

ii. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep, awk

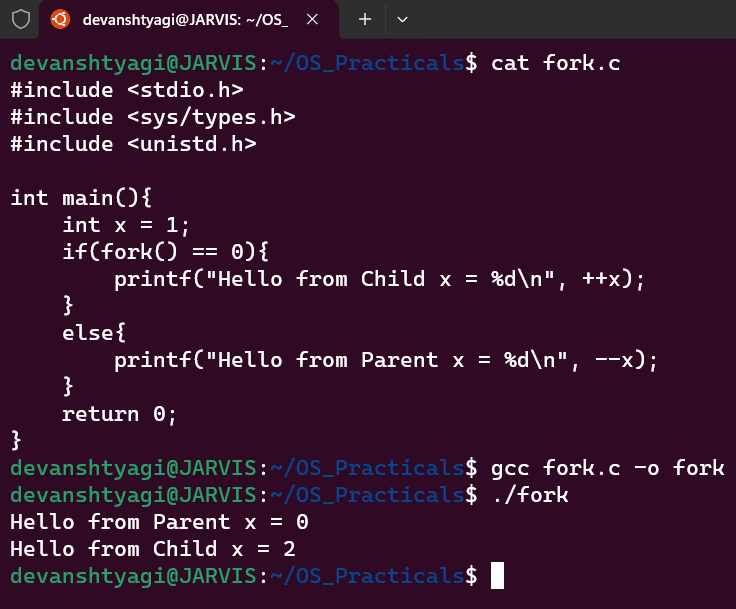
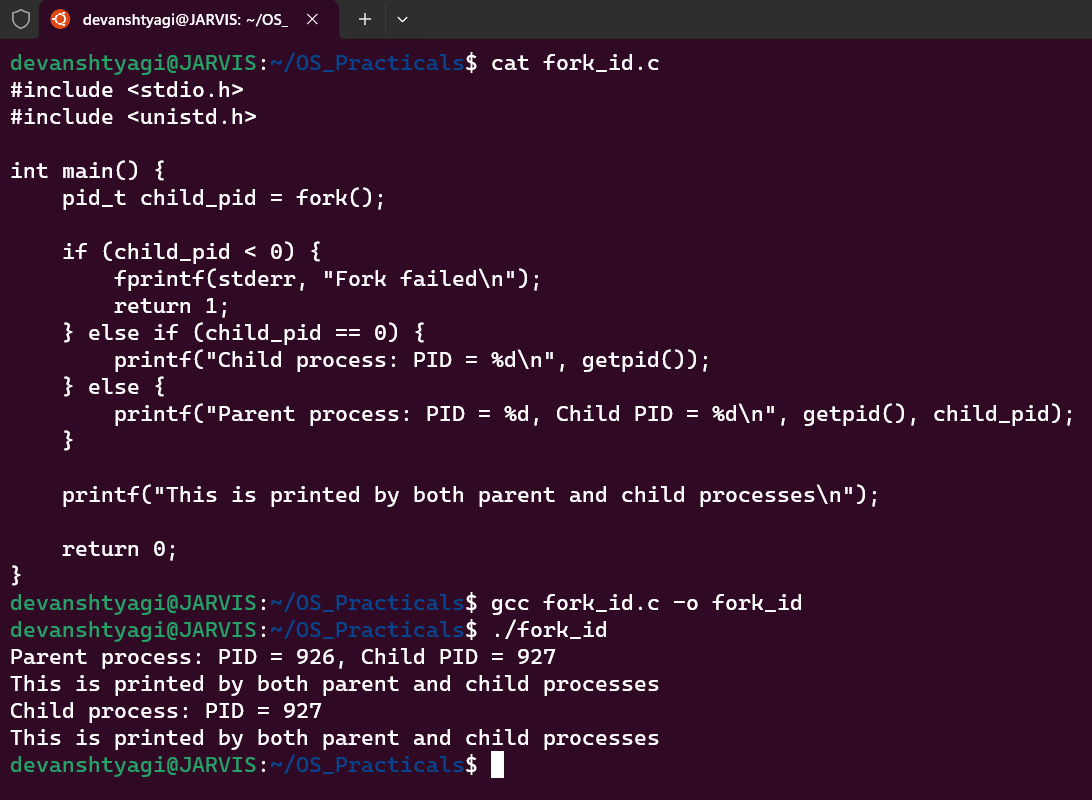
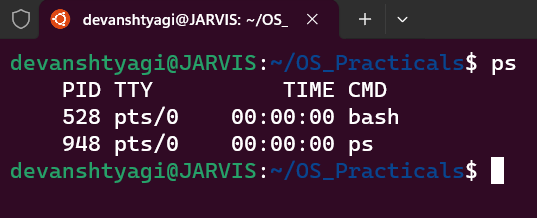
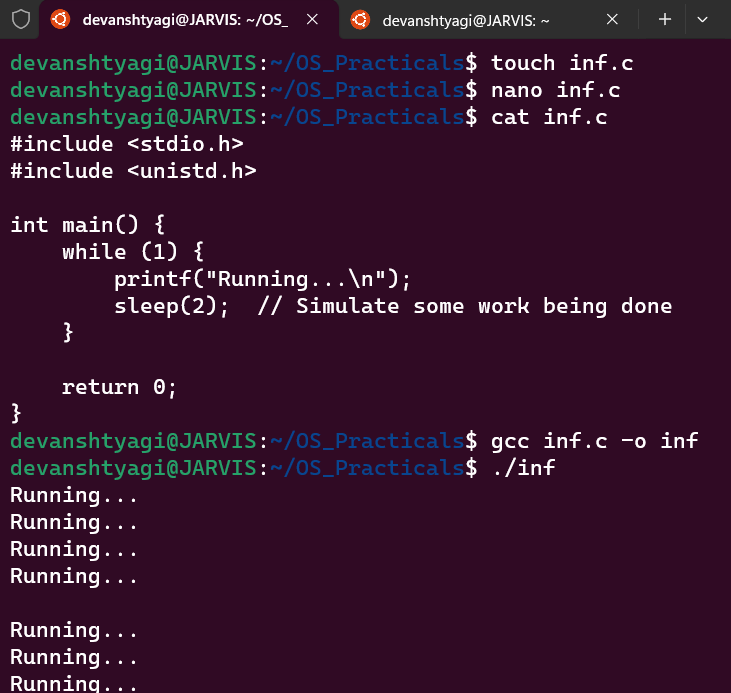
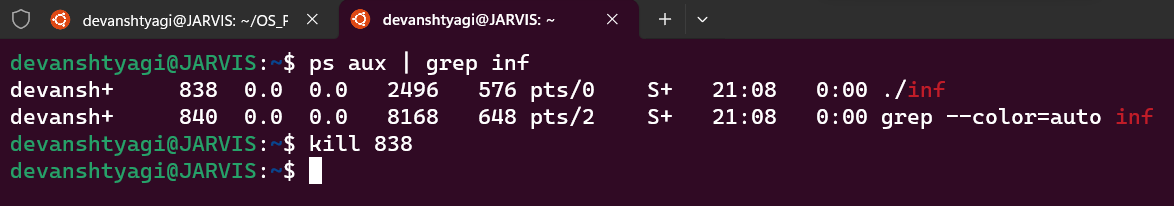
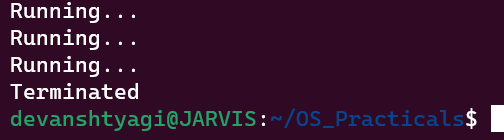
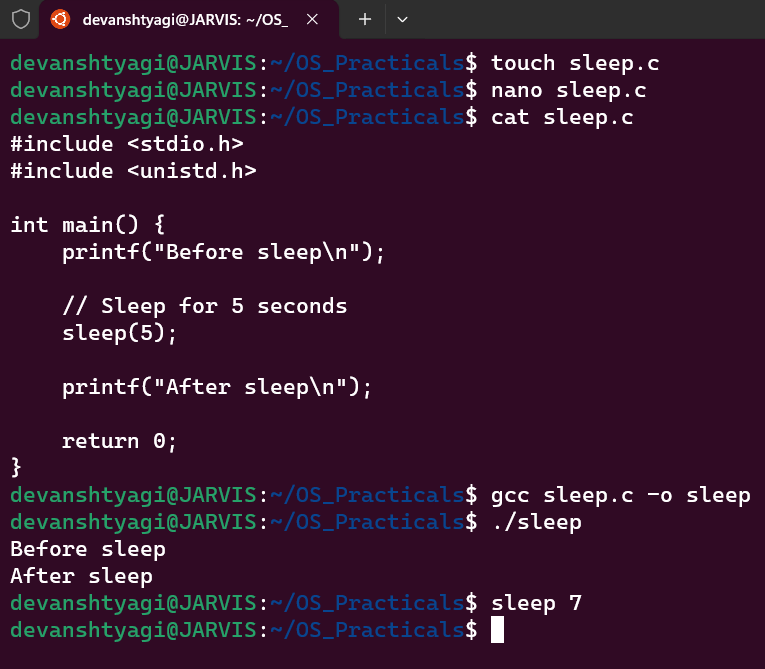
     

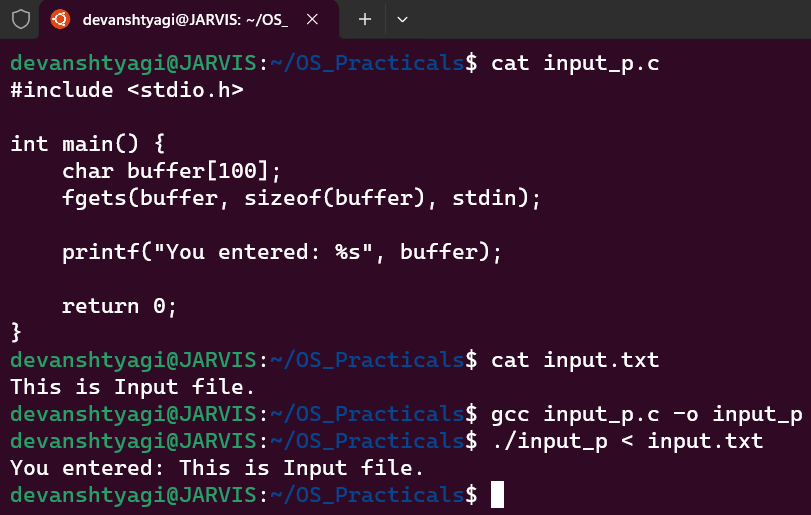
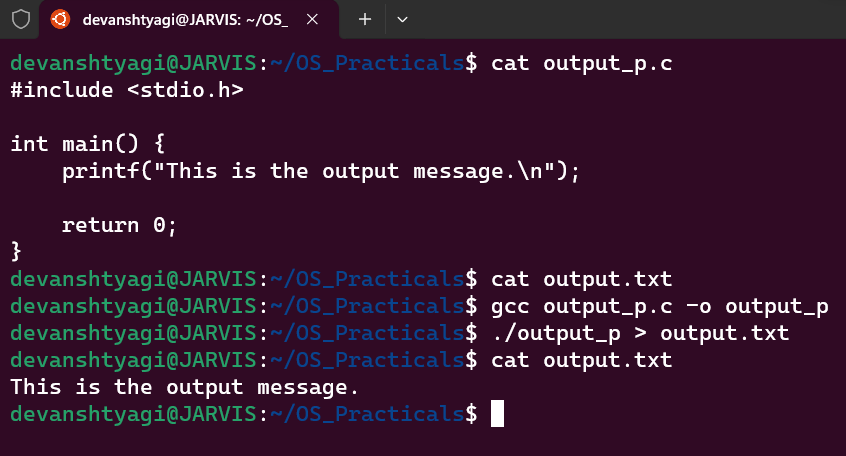
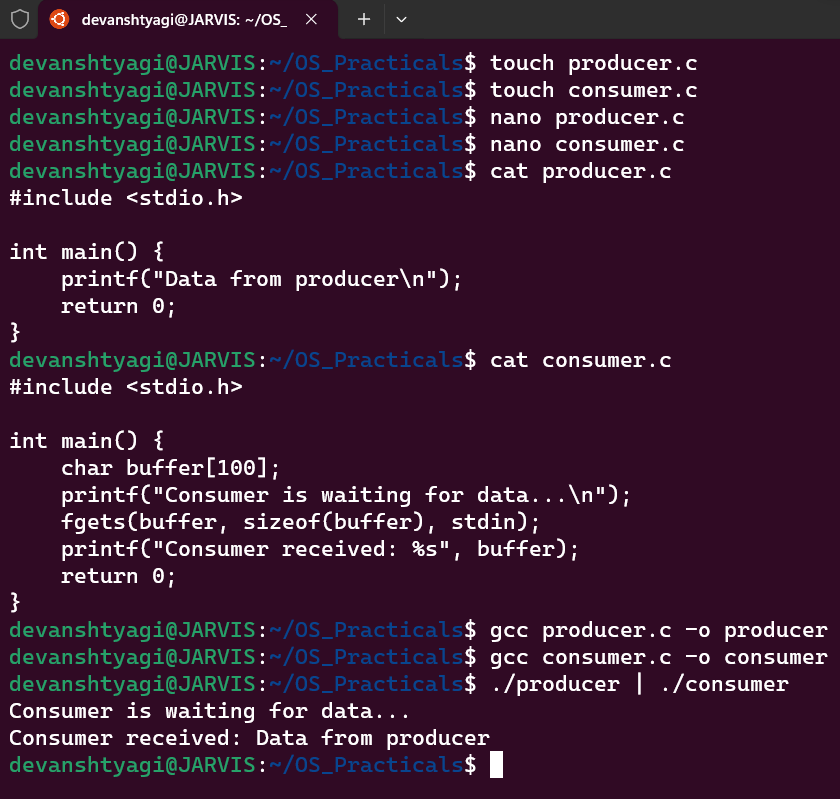
iii. Directory Management : cd, mkdir, rmdir, ls



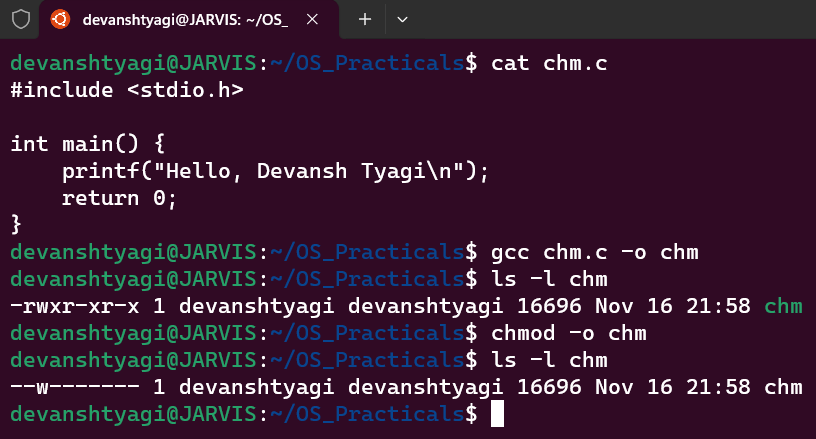
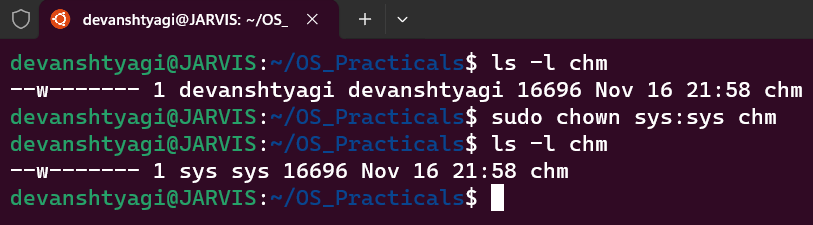
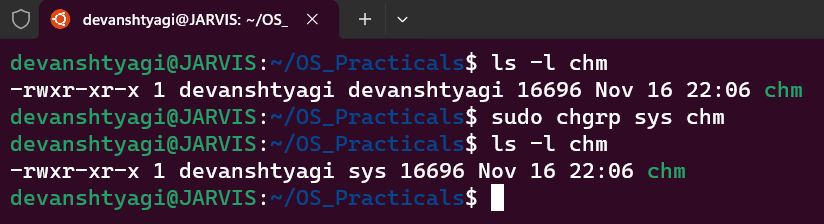
2. Execute various LINUX commands for:

i. Process Control: fork, getpid, ps, kill, sleep

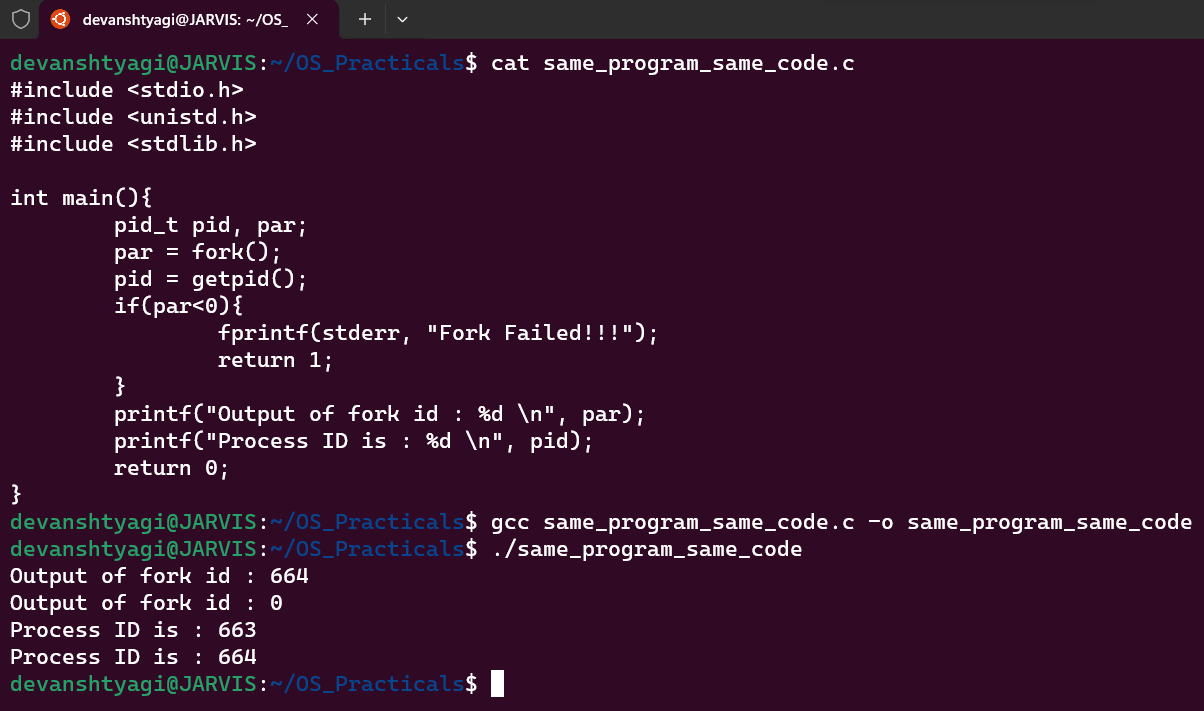
ii. Communication: Input-output redirection, Pipe  

iii. Protection Management: chmod, chown, chgrp

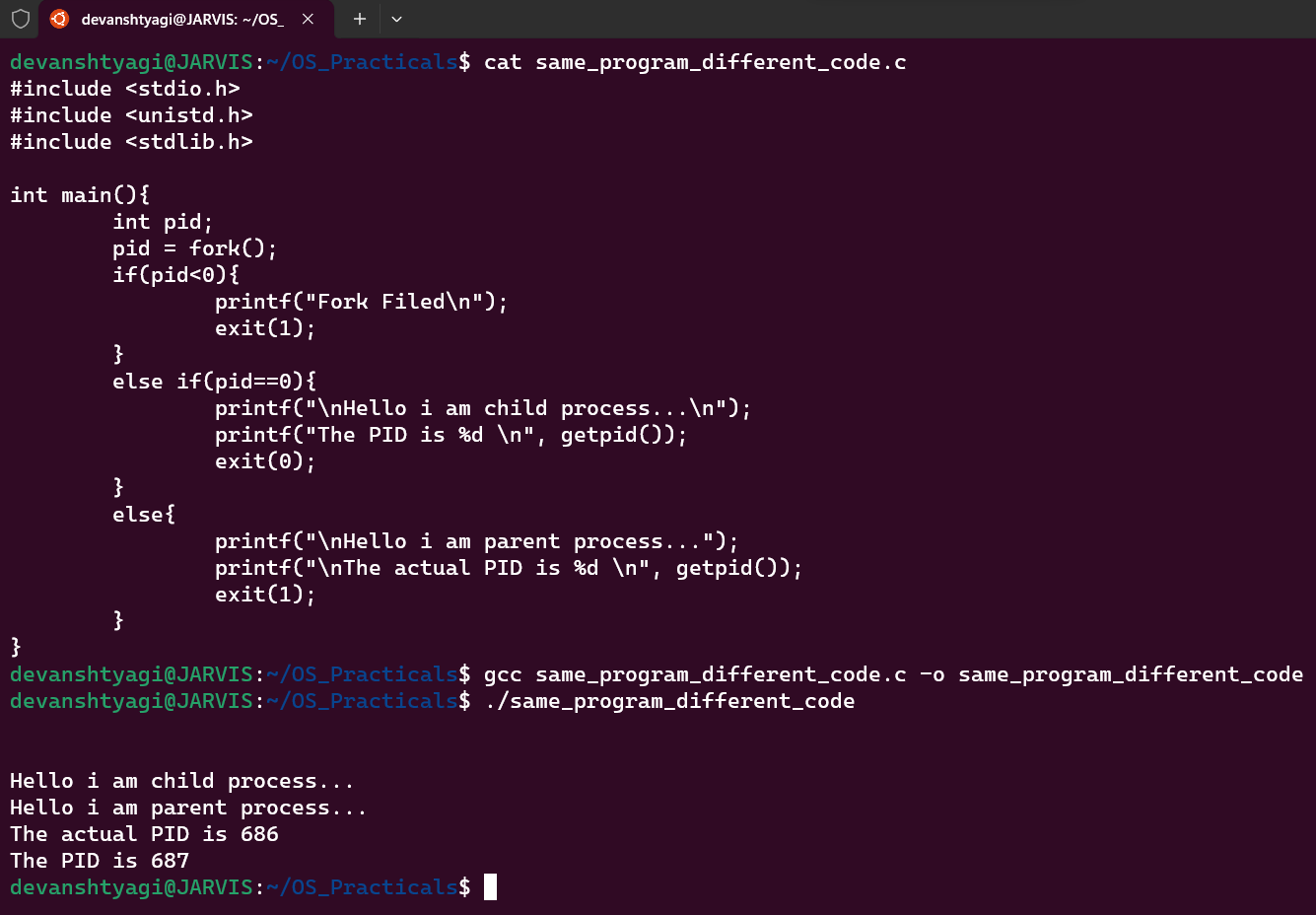
  

3. Write a program(using fork() and/or exec() commands) where parent and child execute:

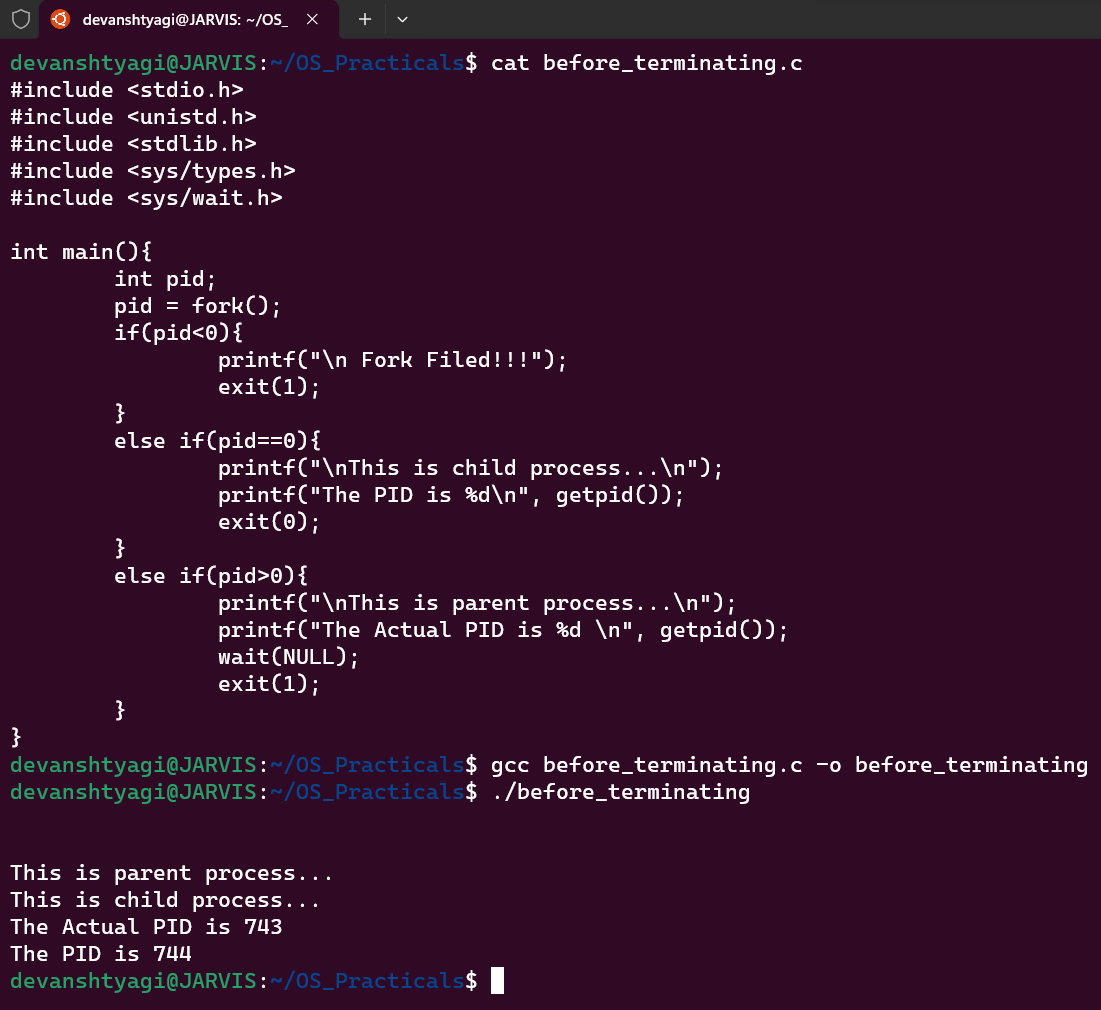
i. same program, same code.



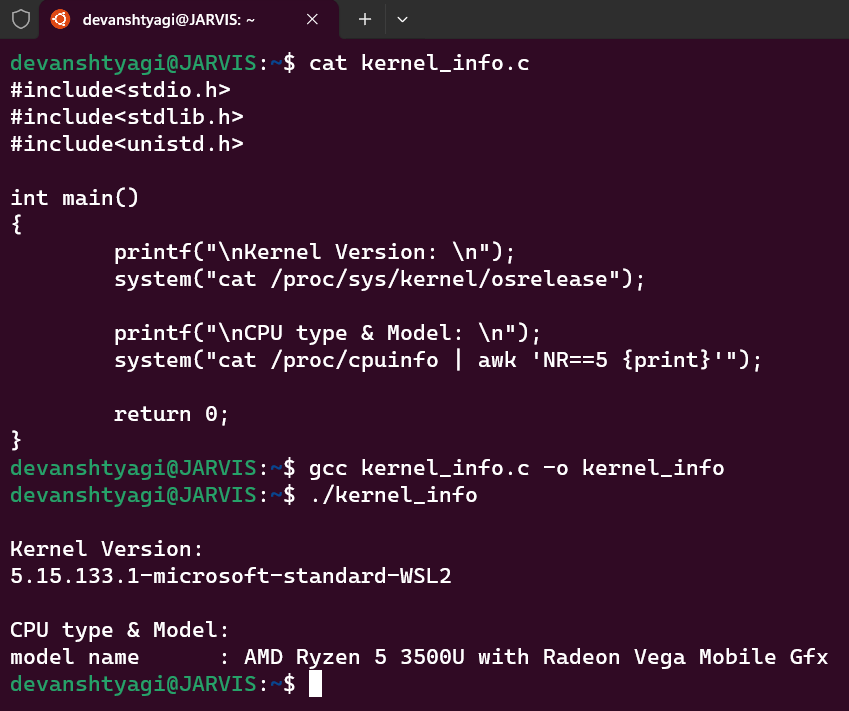
ii. same program, different code.



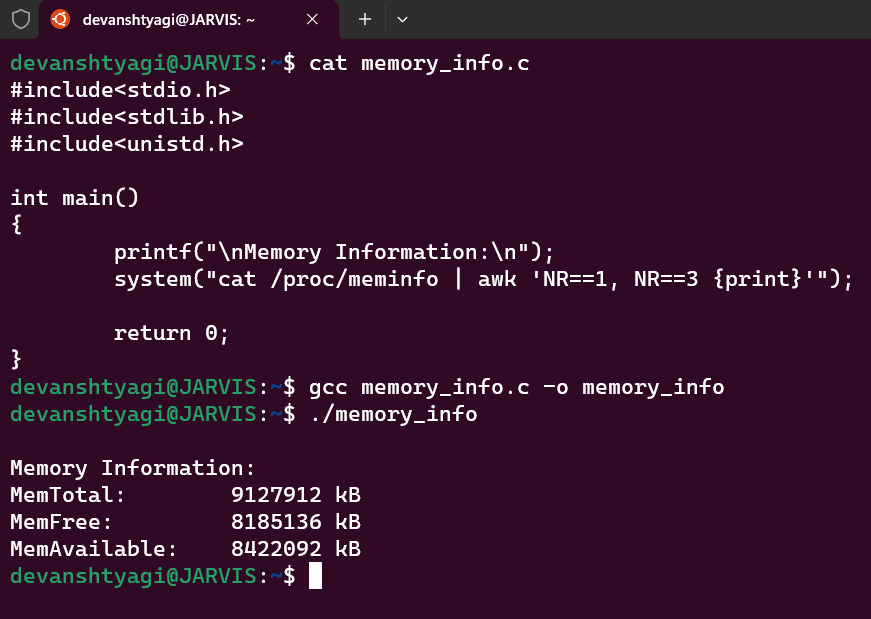
iii. before terminating, the parent waits for the child to finish its task.

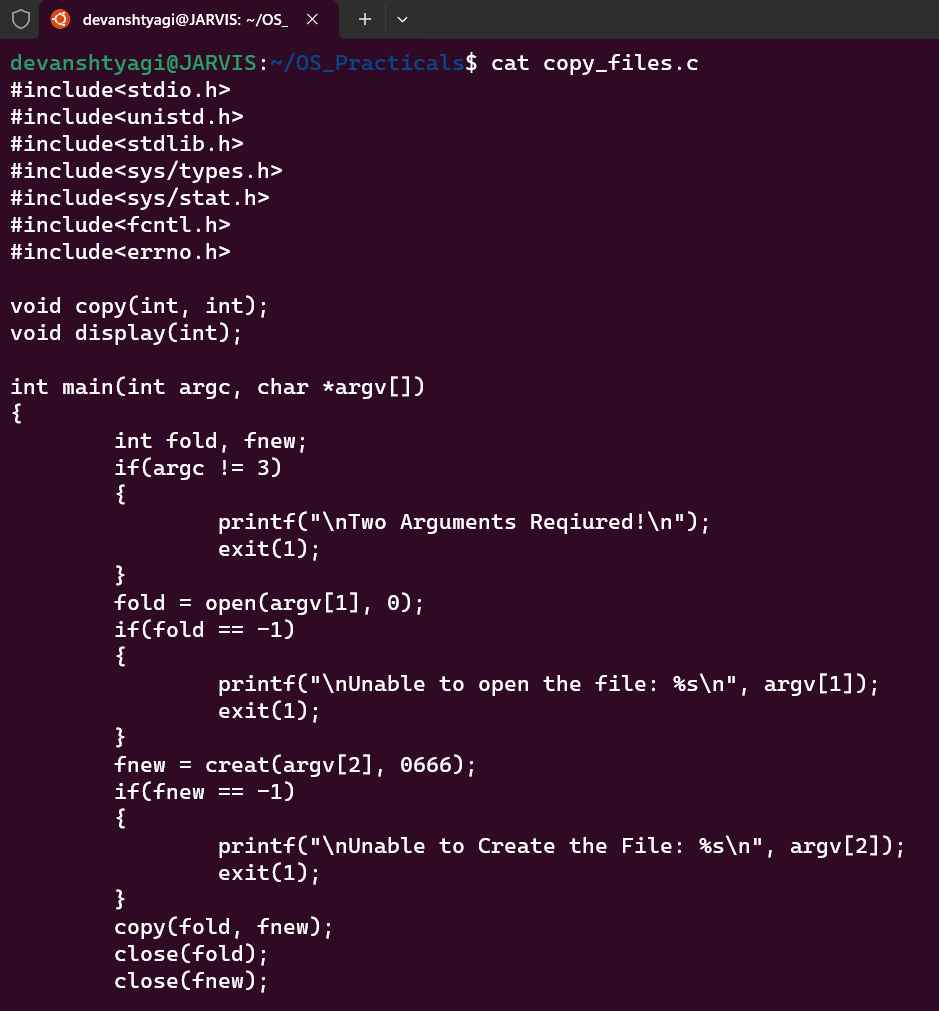
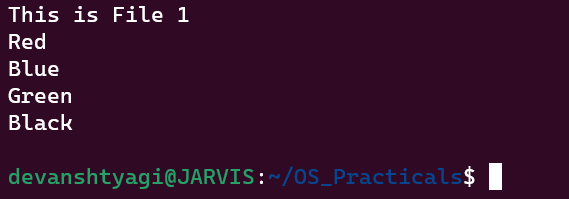


4. Write a program to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)

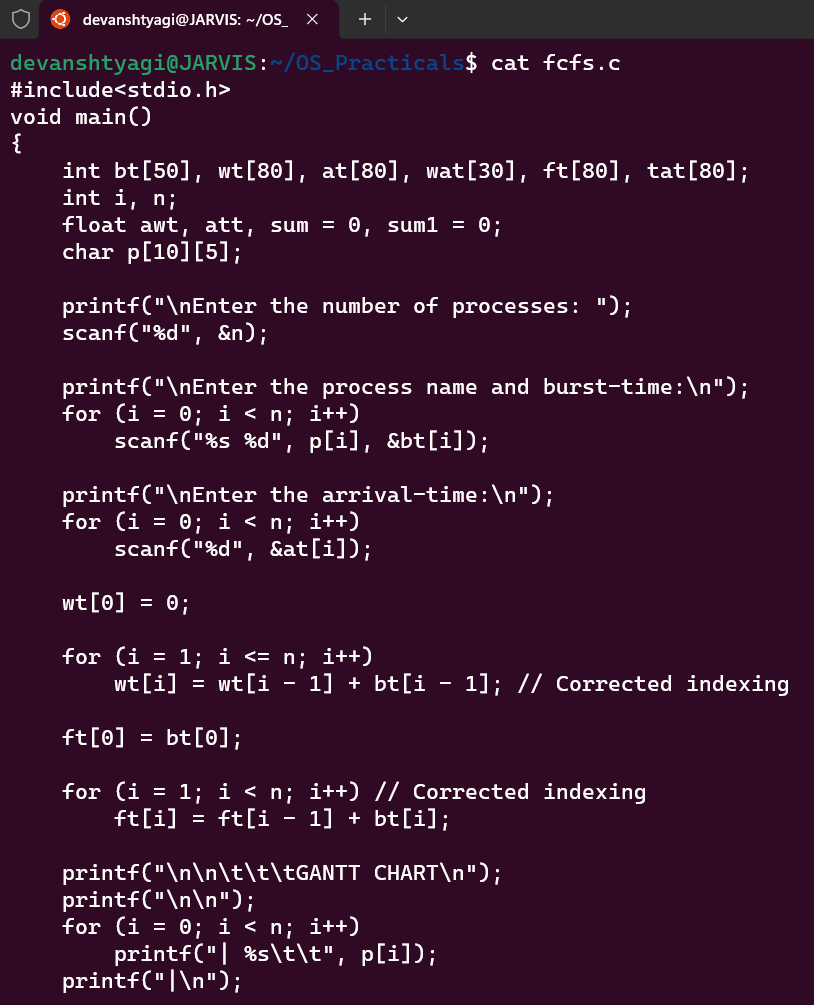
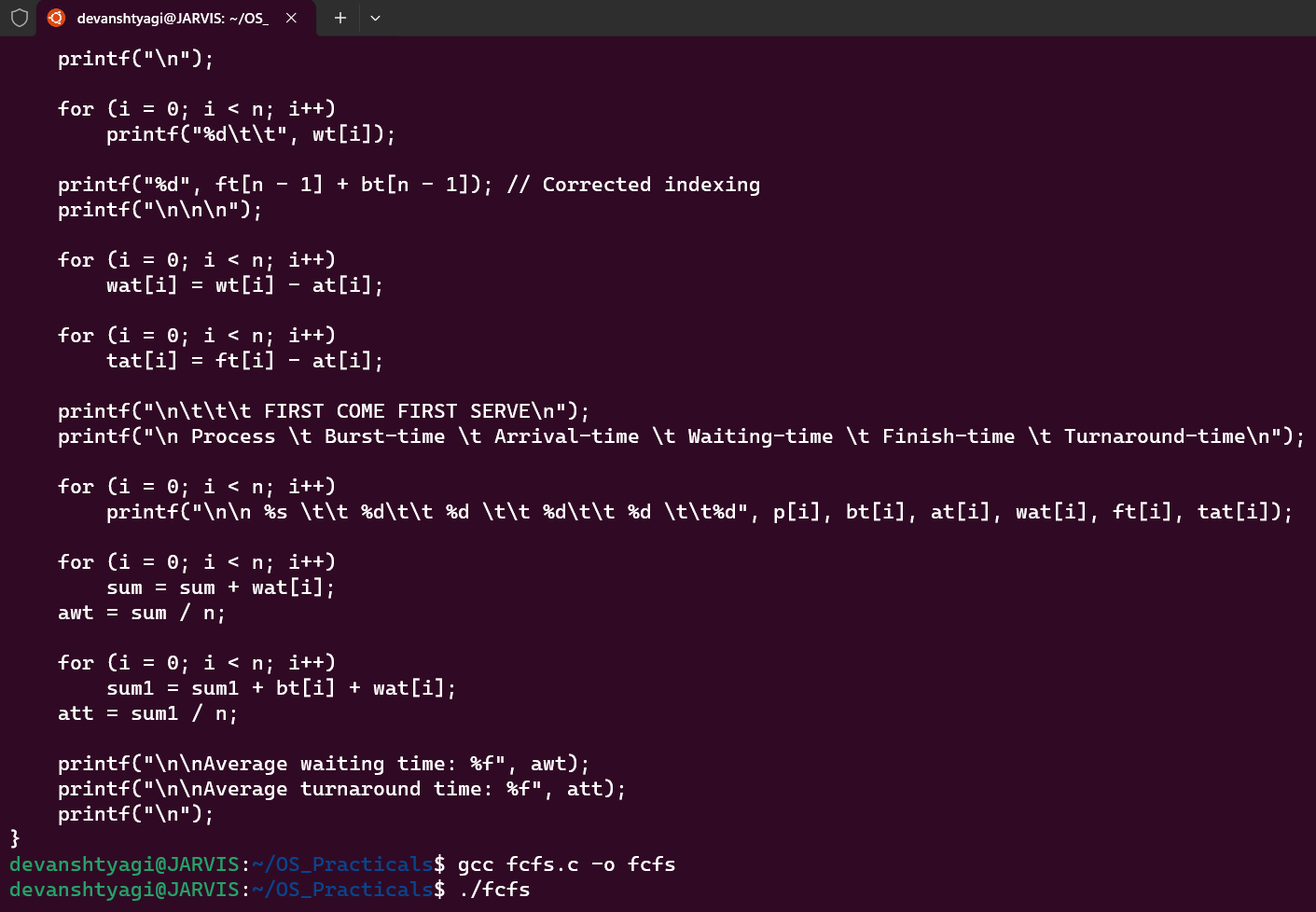
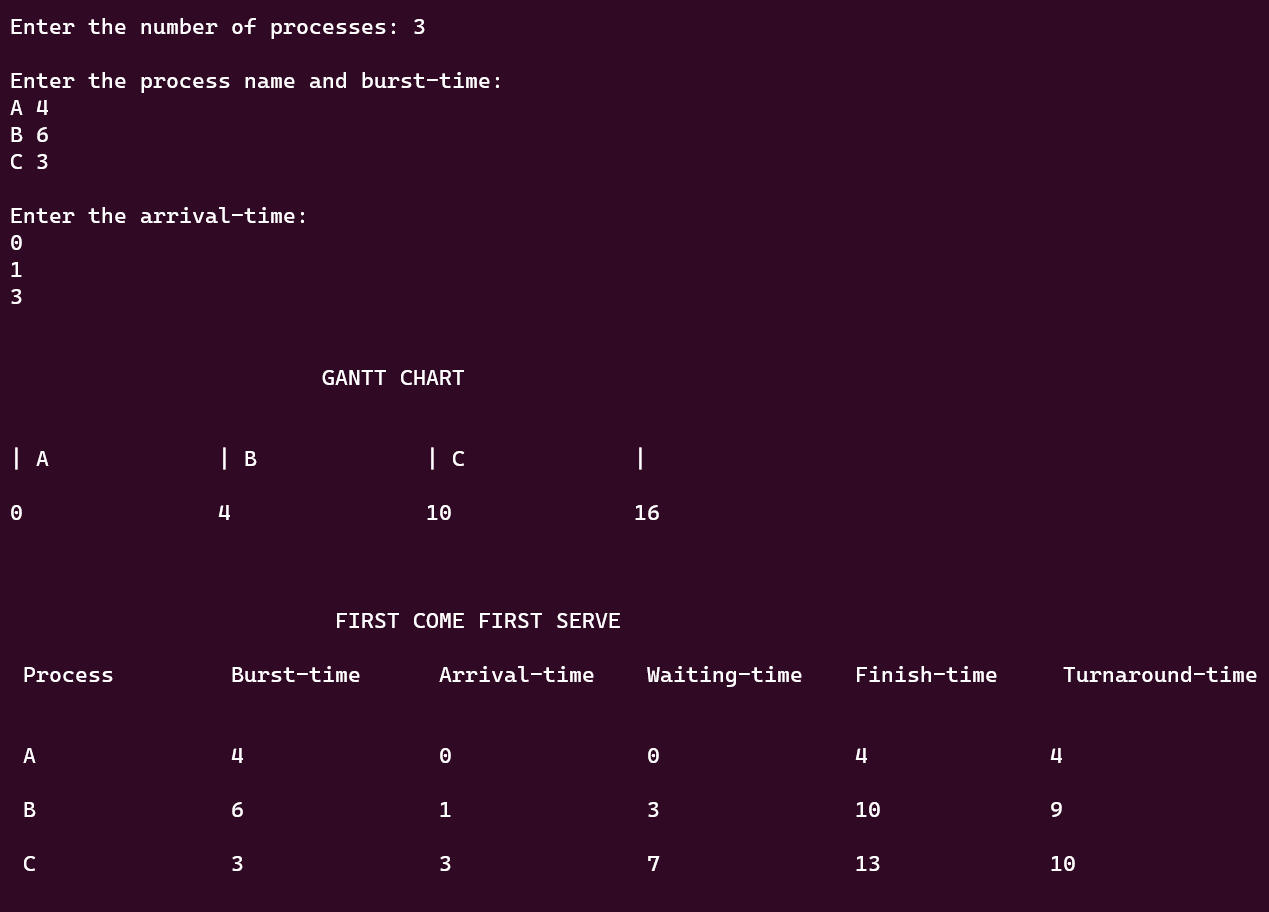
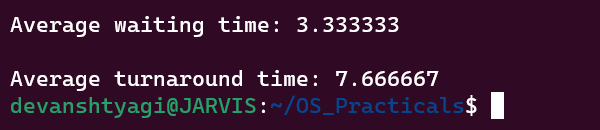


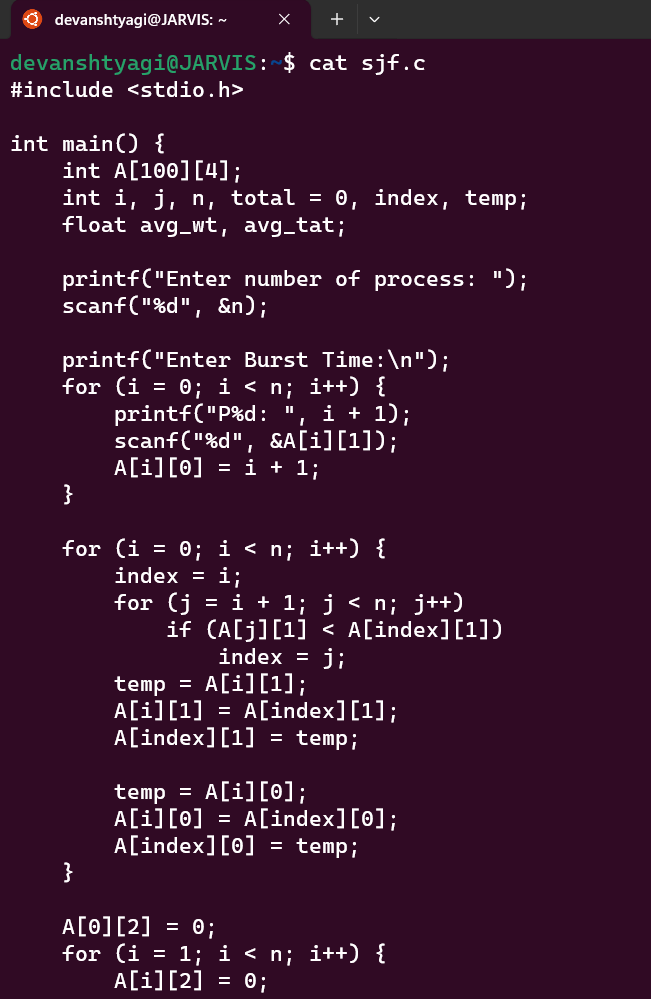
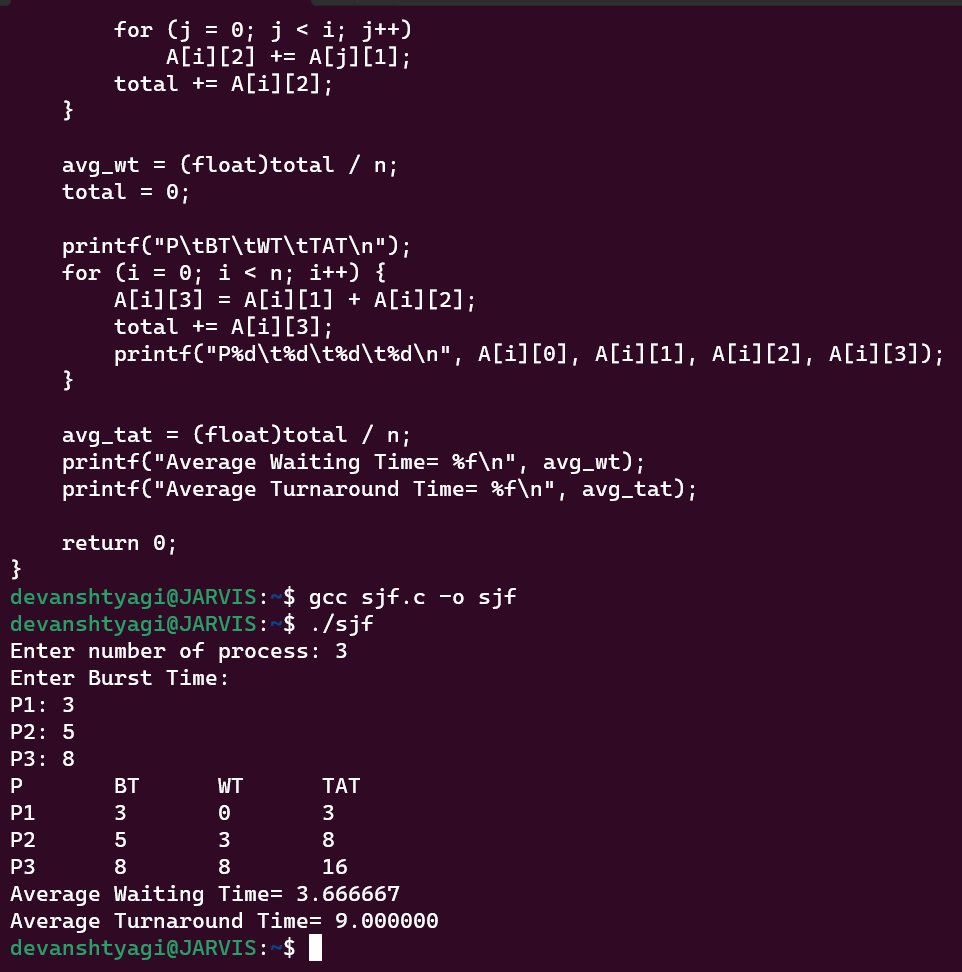
5. Write a program to report behaviour of Linux kernel including information on 19 configured memory, amount of free and used memory. (Memory information)

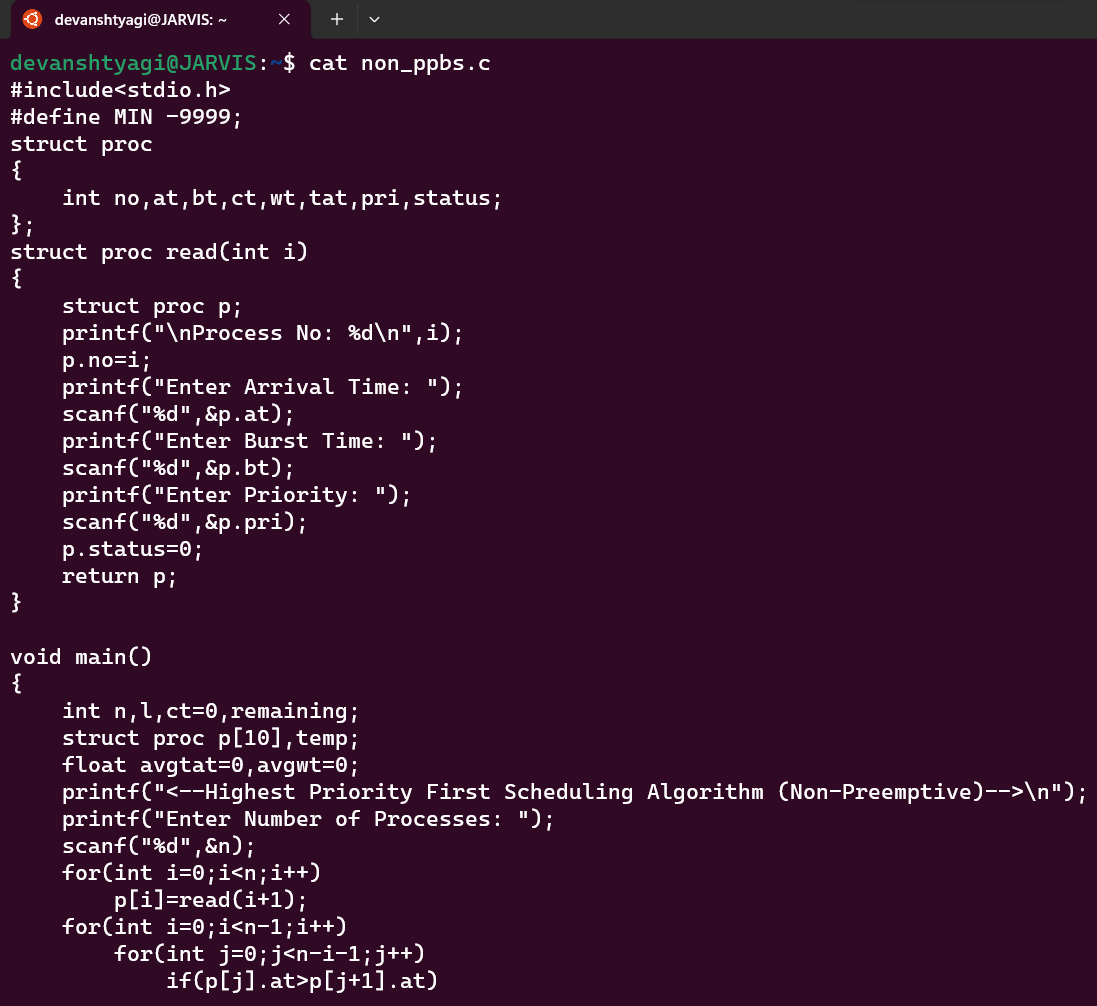
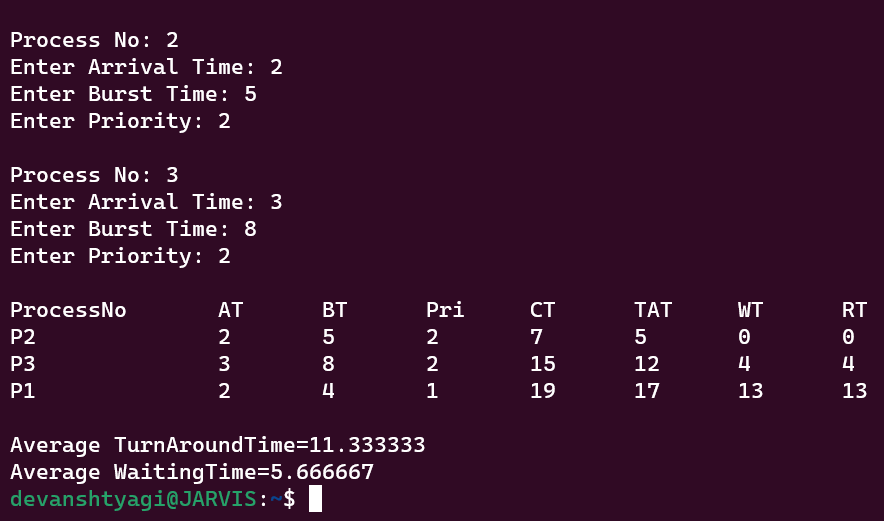


6. Write a program to copy files using system calls.  

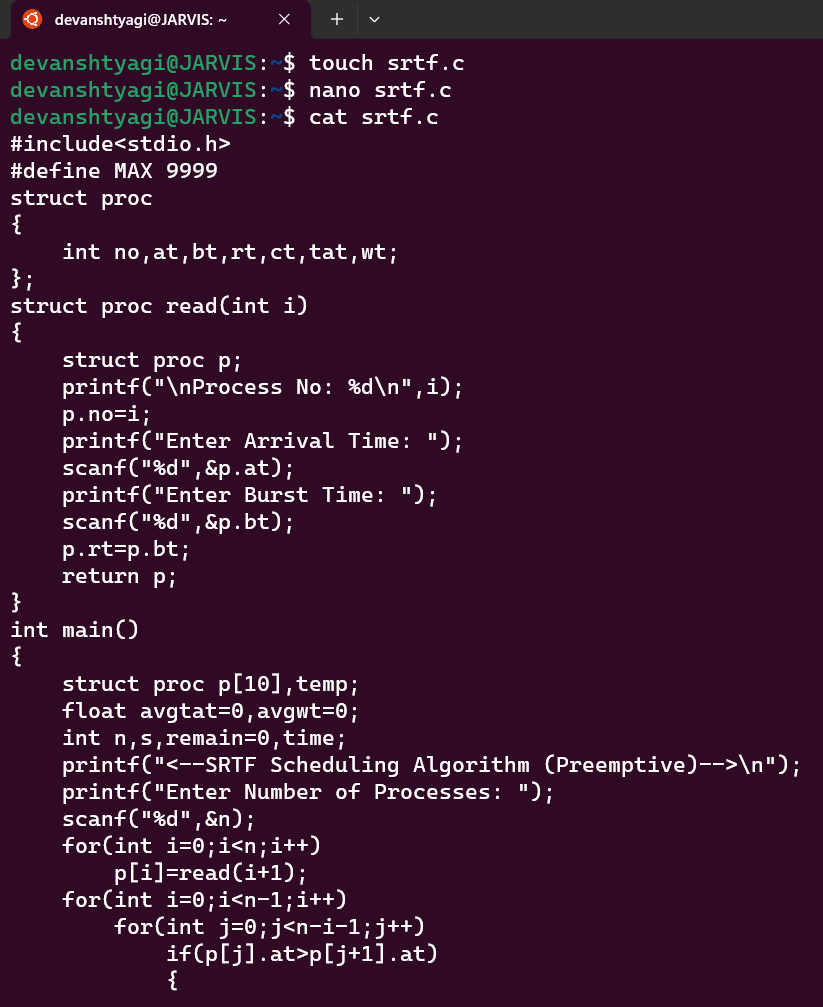
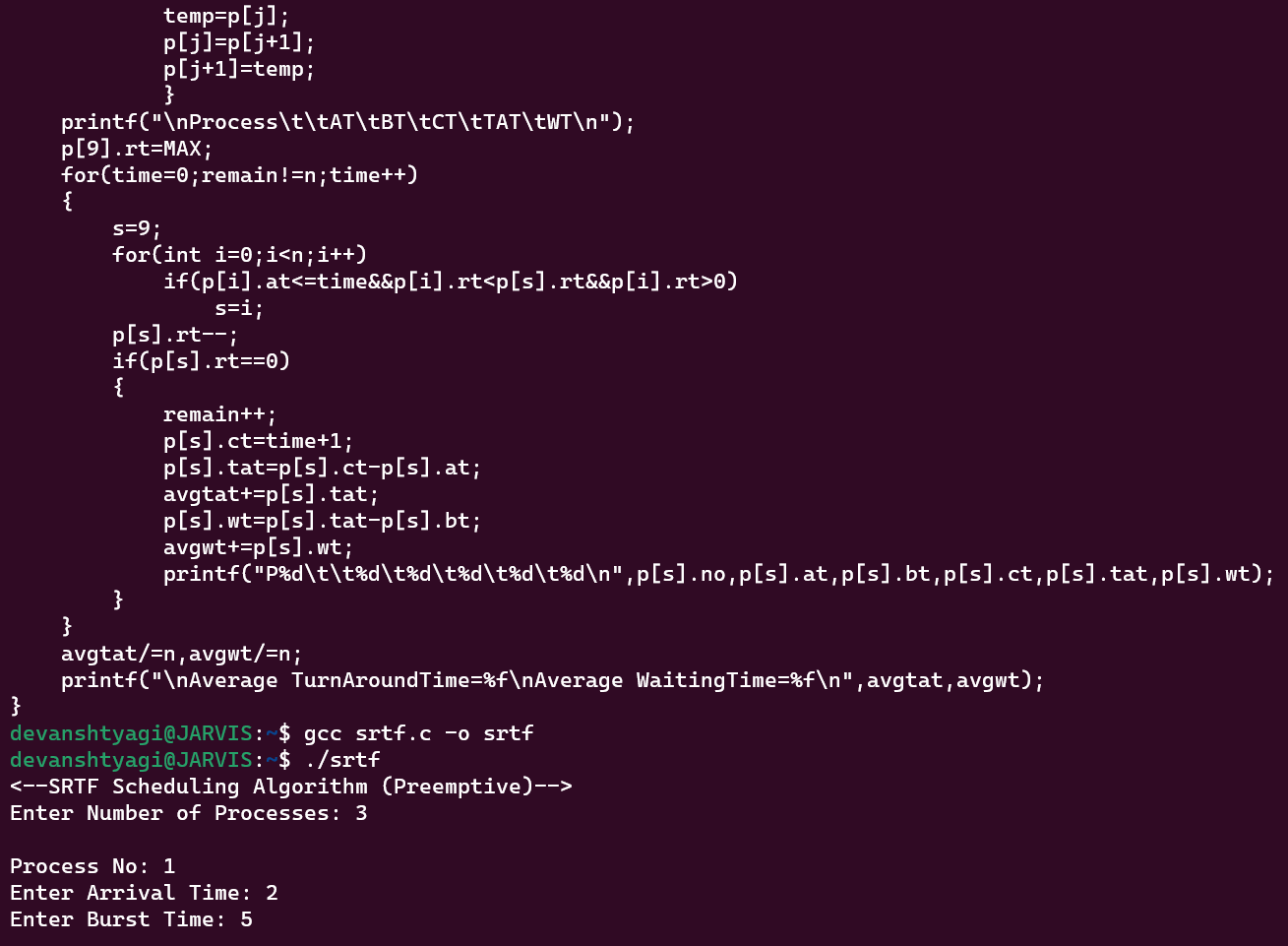
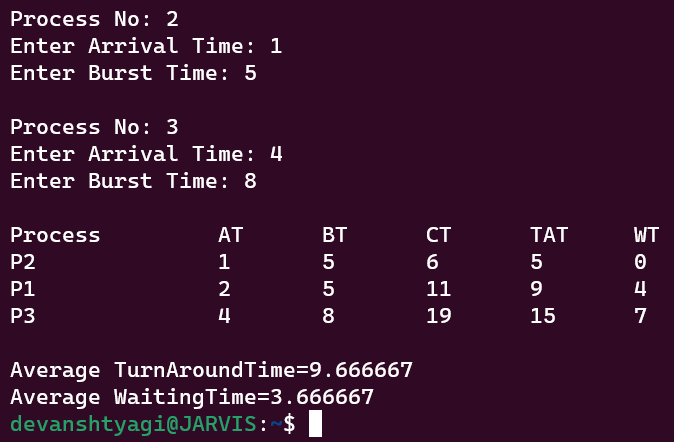
7. Write a program to implement FCFS scheduling algorithm.

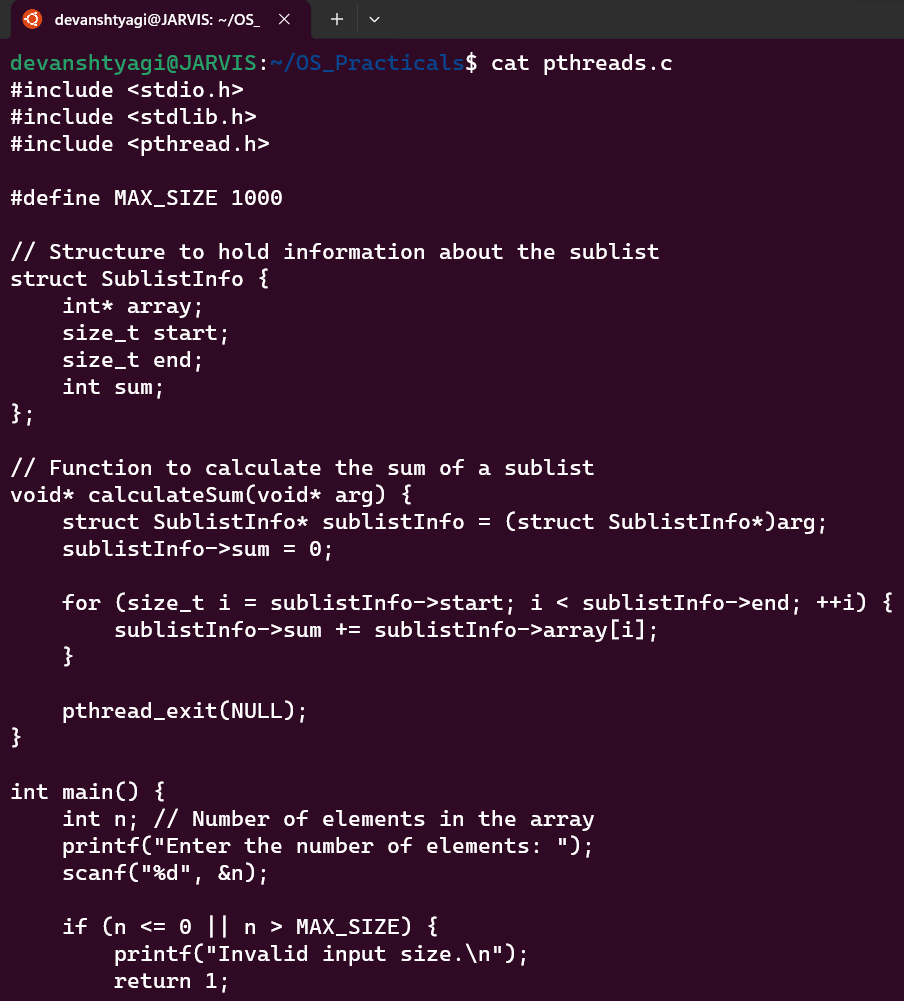
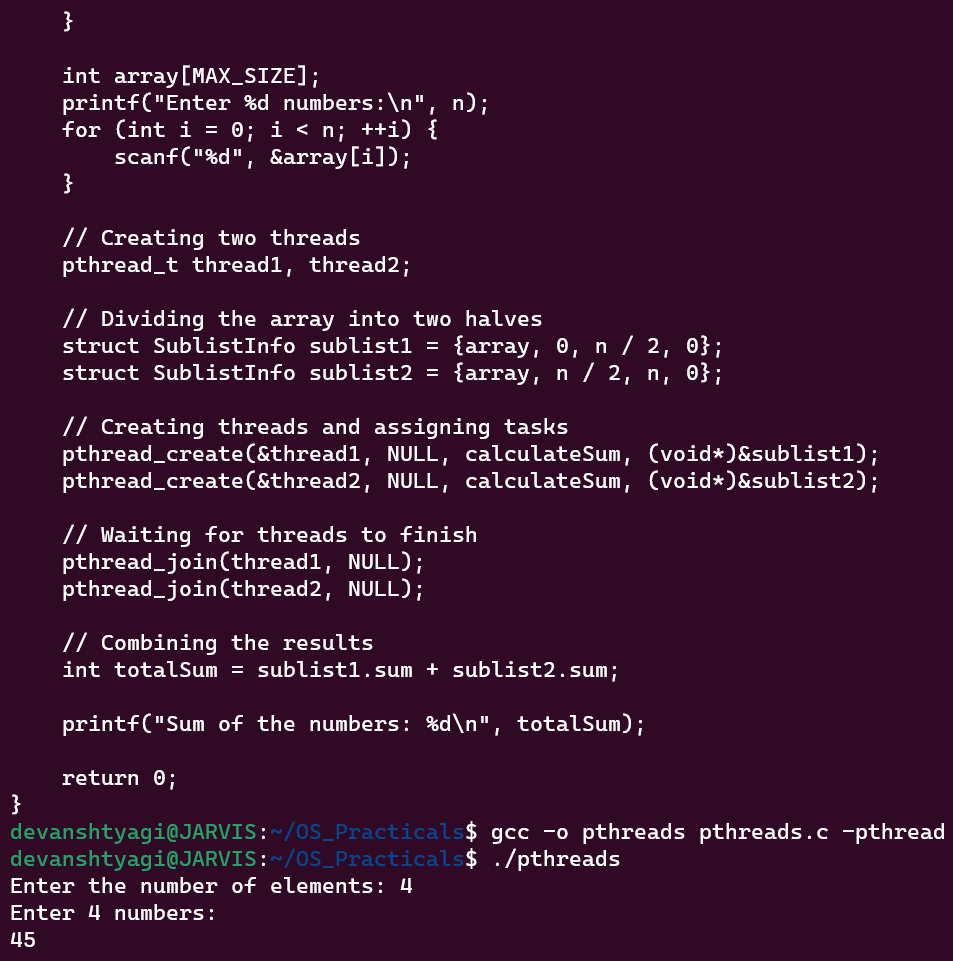
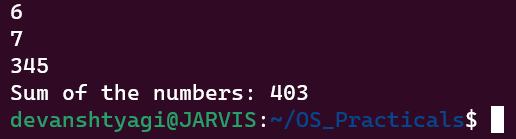
8. Write a program to implement SJF scheduling algorithm 

9. Write a program to implement non-preemptive priority based scheduling algorithm.  

10. Write a program to implement SRTF scheduling algorithm.

11. Write a program to calculate sum of n numbers using Pthreads. A list of n numbers is divided into two smaller list of equal size, two separate threads are used to sum the sublists.

12. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

