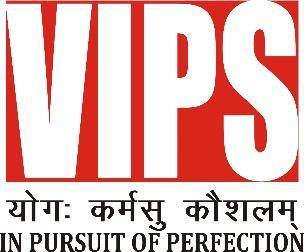
****

**VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS**

**Grade A++ Accredited Institution by NAAC**

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;

Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

**SCHOOL OF ENGINEERING & TECHNOLOGY**

**B. Tech Programme: AI-ML (A)**

**(5th Semester)**

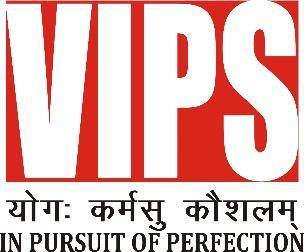
**Course Title: Operating Systems Lab**

**Course Code: AIML- 351**

**Submitted To: Submitted By:**

**Dr. Shivanka Name: Kunsh Sabharwal**

**Enrolment No: 01117711623**

**VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS**

**Grade A++ Accredited Institution by NAAC**

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;

Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

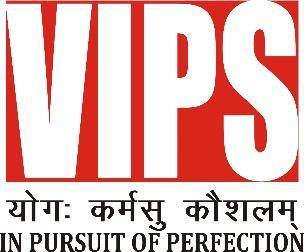
**SCHOOL OF ENGINEERING & TECHNOLOGY**

**VISION OF INSTITUTE**

To be an educational institute that empowers the field of engineering to build a sustainable future by providing quality education with innovative practices that supports people, planet and profit.

**MISSION OF INSTITUTE**

To groom the future engineers by providing value-based education and awakening students' curiosity, nurturing creativity and building  
capabilities to enable them to make significant contributions to the world.

**VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS**

**Grade A++ Accredited Institution by NAAC**

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;

Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

**SCHOOL OF ENGINEERING & TECHNOLOGY**

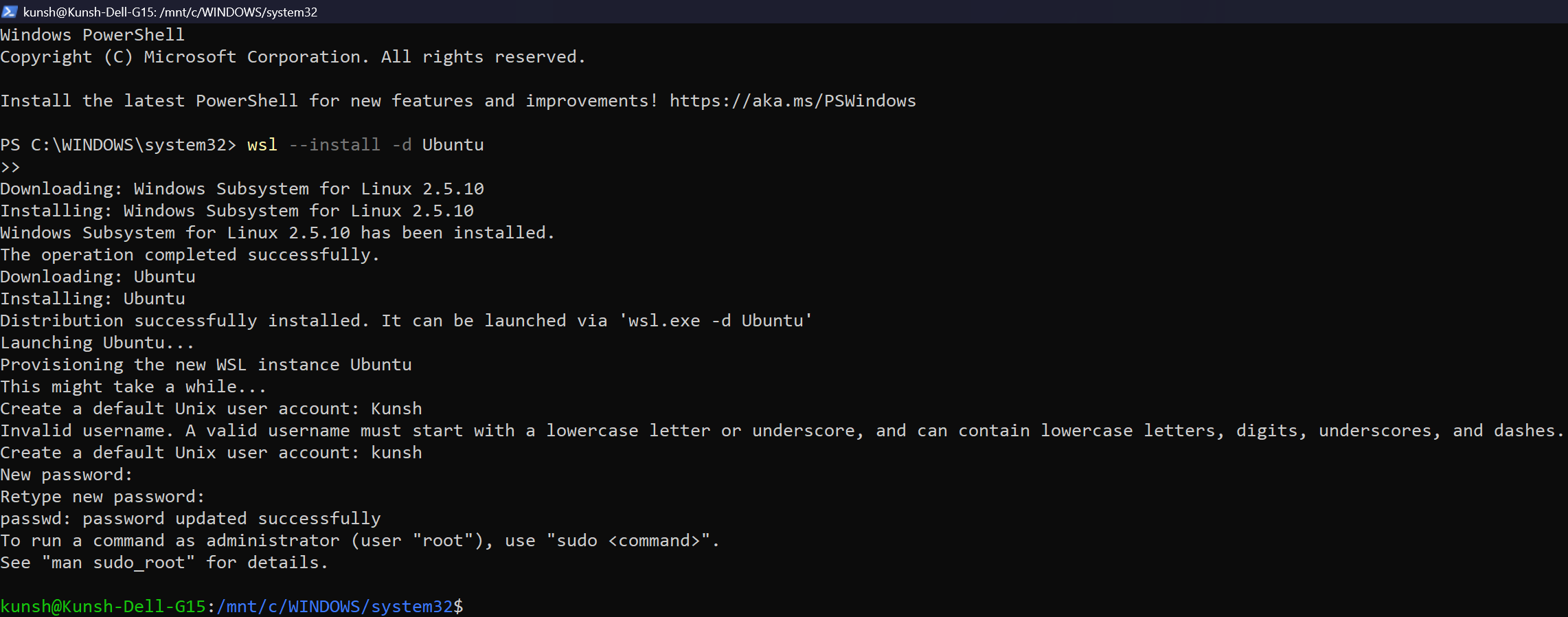
**INDEX**

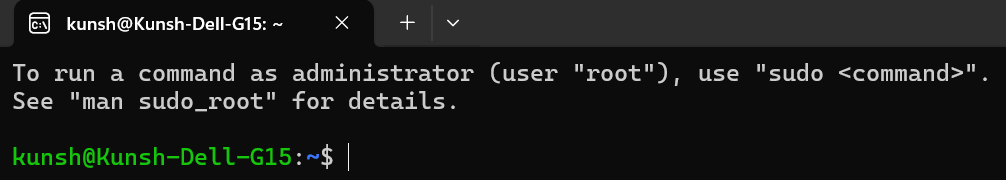
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.No. | Experiment | Date | Marks | | | Remarks | Updated Marks | Faculty Signature |
| Laboratory Assessment (15 Marks) | Class Participation (5 Marks) | Viva (5 Marks) |  |  |  |
| 1. | Install Ubuntu on windows and, compile and run first C program using gcc. |  |  |  |  |  |  |  |
| 2. | Open ubuntu terminal and write the command for following operations and share the output screen  (a) Command to know your current working directory  (b) List all the files in the current directory  (c) List all the files in the order of their file size  (d) List only directories in the current folder  (e) List only files starting with “N” alphabet  (f) Using help command find the help on ‘man’ command.  (g) Display the content of manual pages on ‘ls’ command  (h) Demonstrate the usage of “whatis” command.  (i) Make directory named “OSLab/kunsh”.  (j) Write command to reach to “kunsh” Directory.  (k) Create a txt file named “TodaysMsg.txt” and write a greeting message in it.  (k) Copy this file “TodaysMsg.txt” to OSLab directory  (l) Delete the file “TodaysMsg.txt” from the ‘yourname’ Folder  (m) Delete the directory ‘yourname’  (n) Create a text file named “Hello.txt” and write a suitable message in it.  (o) Using touch command create files with names mon.txt, tues.txt, and wed.txt  (p) Copy these newly created files to a folder named “dupfolder” after creating it.  (q) Move Hello.txt to dupfolder  (r) Count number of words in the Hello.txt file |  |  |  |  |  |  |  |
| 3. | Perform following shell script-based programs  (a) Write a Shell Program to swap the two integers.  (b) Create a shell script that checks if a specific directory exists. If it does, the script should back up all files from that directory into a specified backup directory. The script should then loop through the files in the backup directory and list all files that were successfully copied. If the directory does not exist, the script should print an error message.  (c) Write a shell script to check if a given number is a prime number or not  (d) Write a shell script to greet the user as per the time whenever he/ she opens terminal. |  |  |  |  |  |  |  |
| 4. | Write a c program to implement the following scheduling algorithms.  (a) First come first serve  (b) Round Robin Scheduling  (c) Shortest job first  (d) Shortest Job remaining first. |  |  |  |  |  |  |  |
| 5. | Implementation of the following Memory Allocation Methods for fixed partition  (a) First Fit  (b) Worst Fit  (c) Best Fit. |  |  |  |  |  |  |  |
| 6. | Write a program to implement reader/writer problems using semaphore. |  |  |  |  |  |  |  |
| 7. | Write a program to implement Banker’s algorithm for deadlock avoidance. |  |  |  |  |  |  |  |
| 8. | Process Management (a) fork()  (b) execv()  (c) execlp()  (d) wait()  (e) sleep()  (A) Program to implement the fork function using C.  (B) Program to implement execv function using C.  (C) Program to implement execlp function.  (D) Program to implement wait function using C.  (E) Program to implement sleep function using C |  |  |  |  |  |  |  |
| 9. | Write a program to implement Inter Process Communication (IPC) using Message Queues. |  |  |  |  |  |  |  |
| 10. | Write a program to implement IPC using pipes. |  |  |  |  |  |  |  |
| 11. | Write a program using Pthread, where main thread calculates number of lines in a file and child calculates number of words. |  |  |  |  |  |  |  |

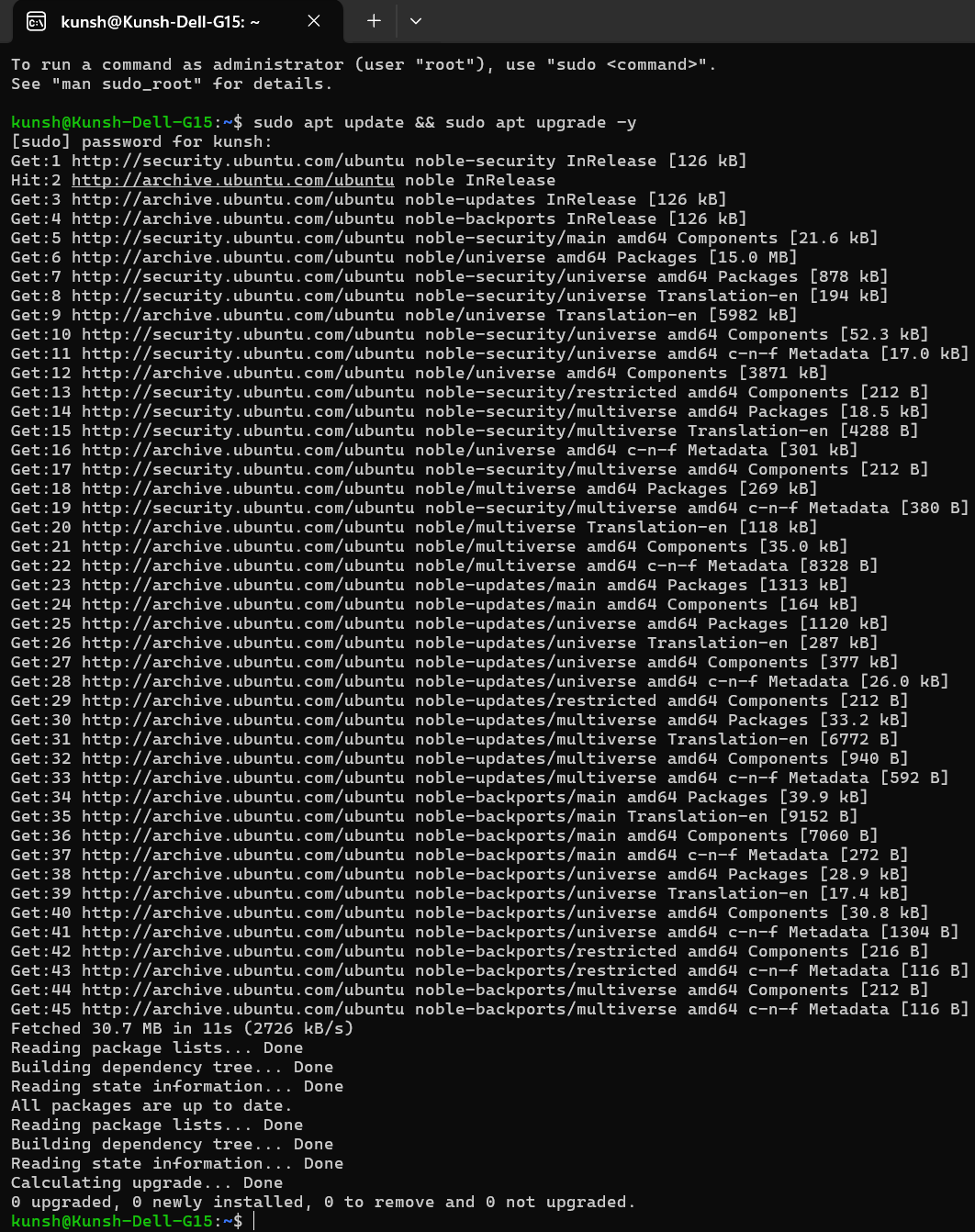
**EXPERIMENT 1**

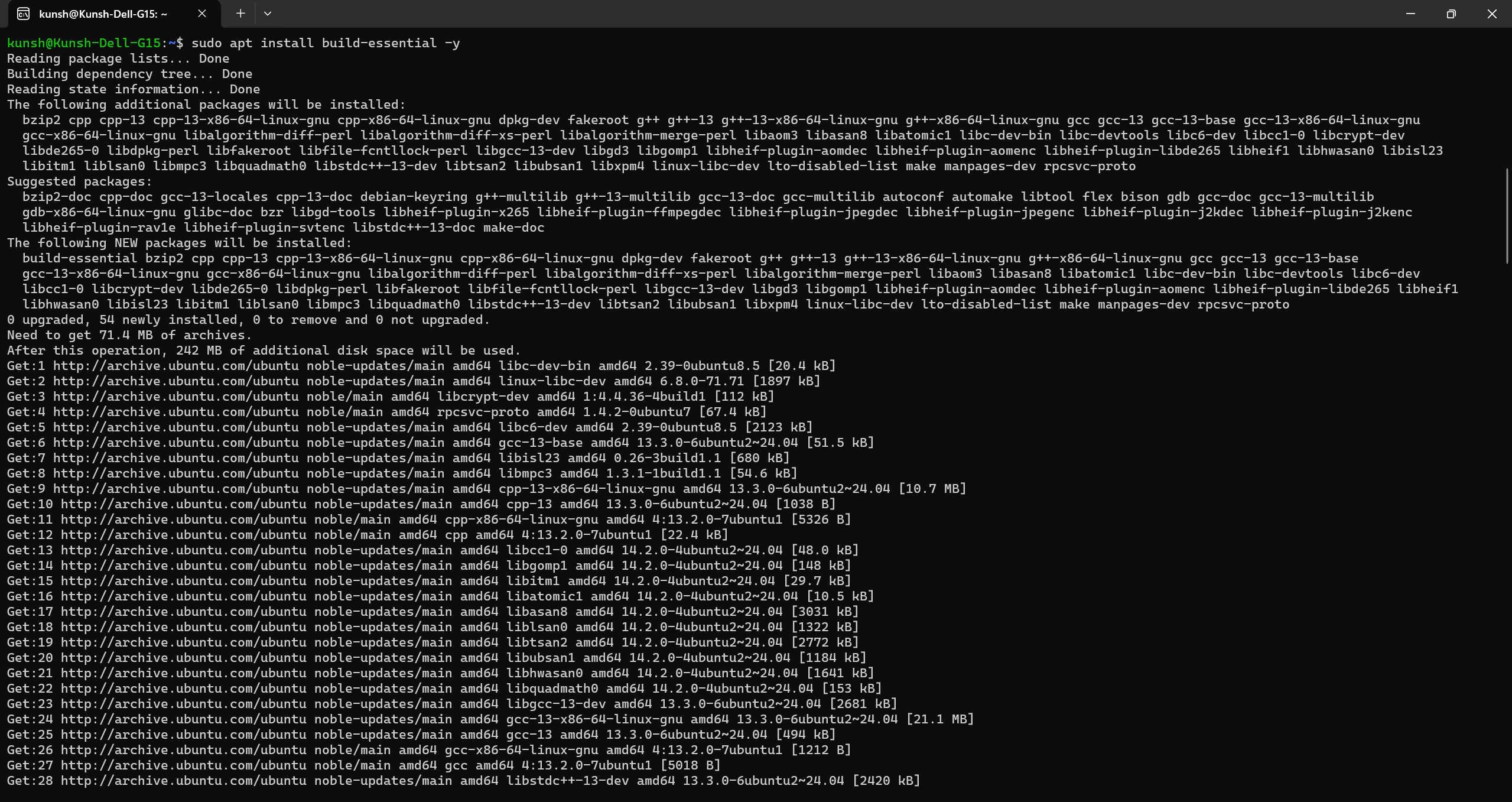
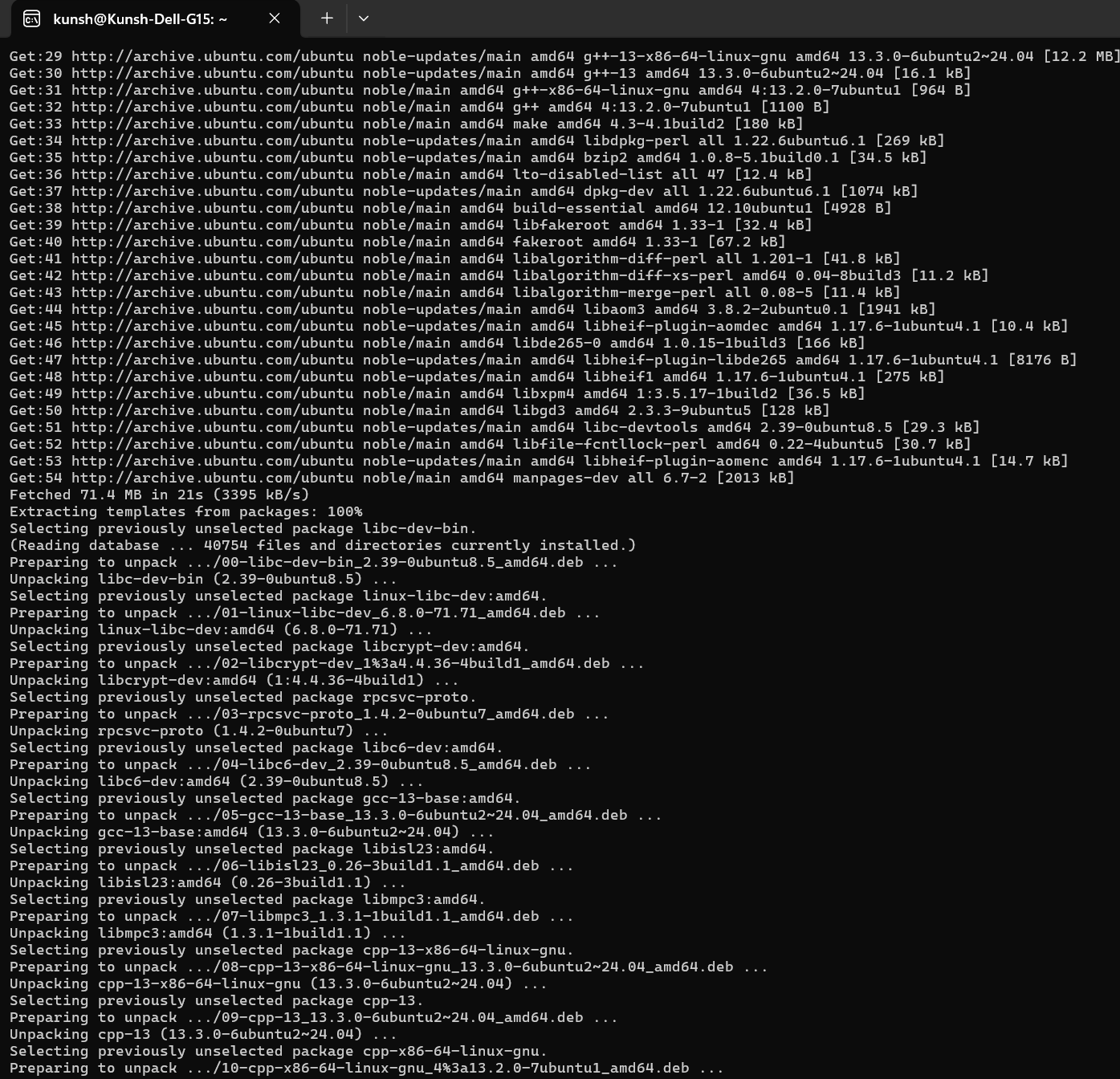
**Problem statement:** Install Ubuntu on windows and, compile and run first C program using gcc.

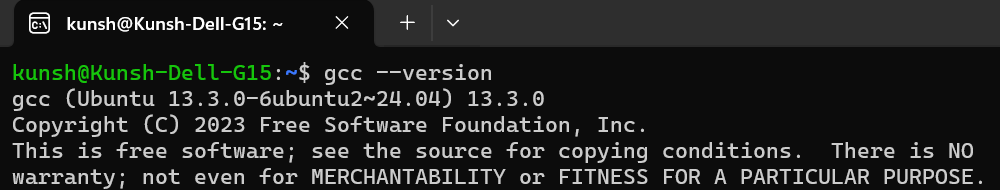
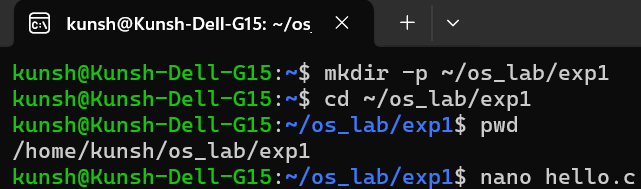
**Theory:**

**Outputs:** (a) Installation Screenshot

(b) Ubuntu Terminal first launch screenshot

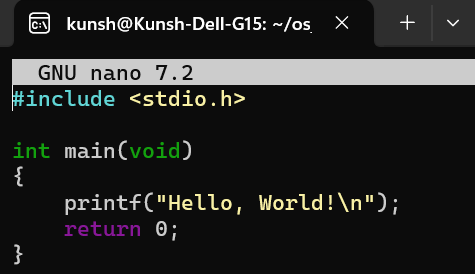
(c) Running Update commands for Ubuntu

(d) Installing gnu and C Compiler

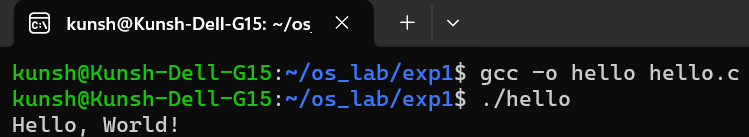
(e) GCC version screenshot

(f) Make new directory and go to it.

(g) Make a new file with name hello.c in nano text editor



(h) Compiling the program and running to see the output

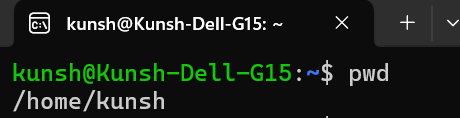
**Learning Outcome:**

**EXPERIMENT 2**

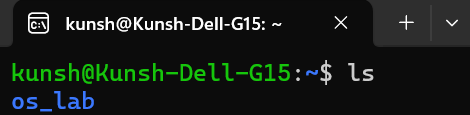
**Problem statement:** Open ubuntu terminal and write the command for following operations and share the output screen.

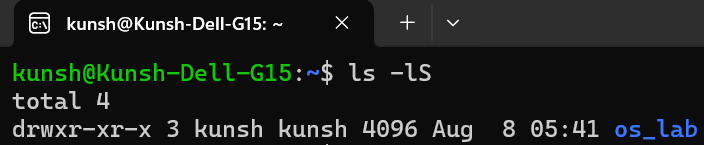
**Theory:**

**Outputs:** (a) Command to know current working directory:

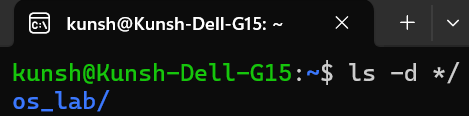


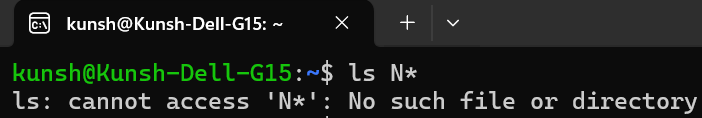
(b) List all the files in the current directory:



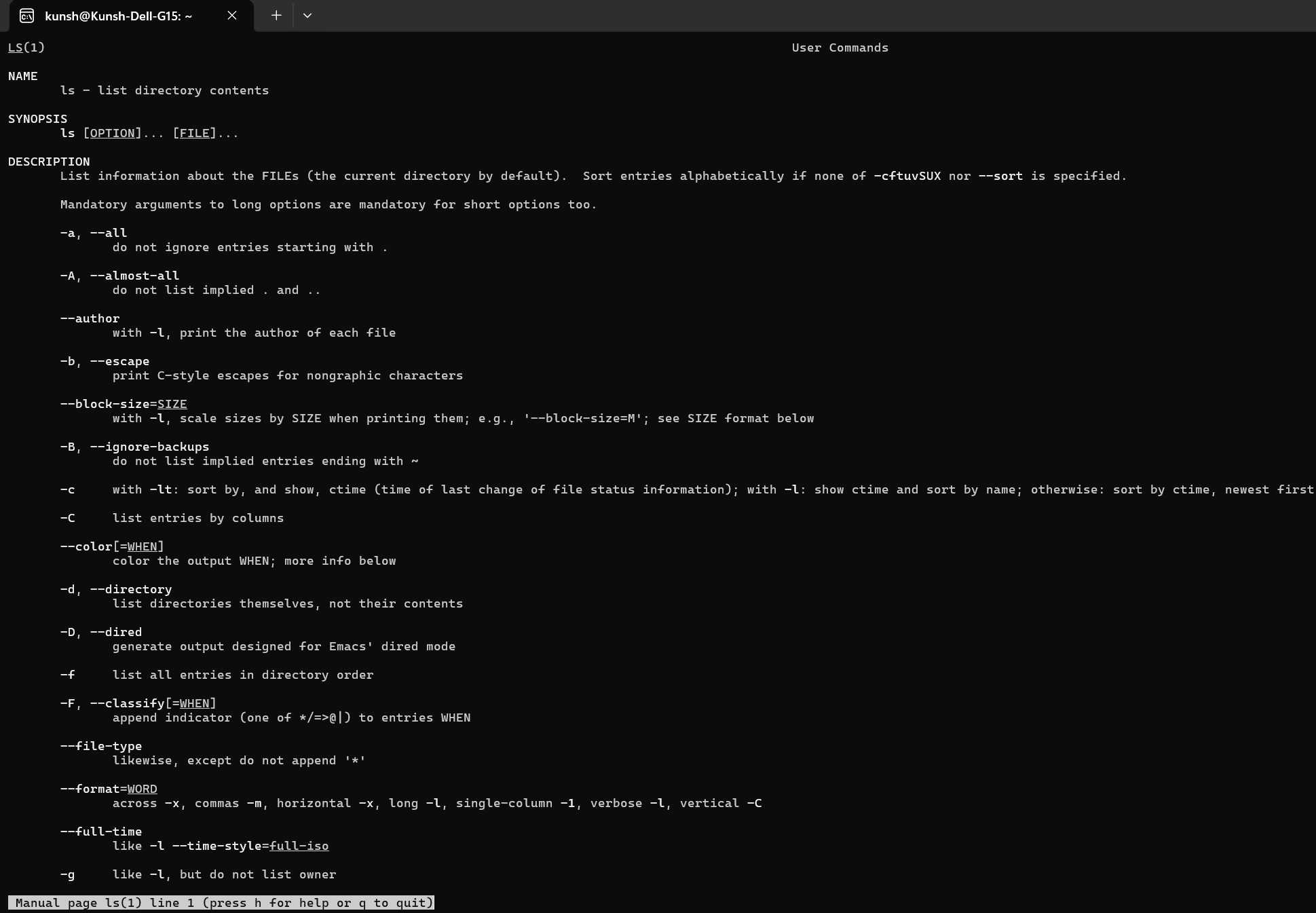
(c) List all files in order of their file size:

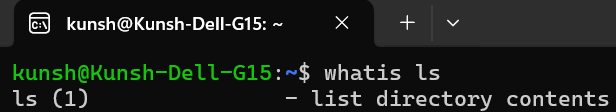
(d) List only directories in the current folder:

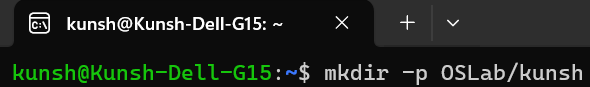


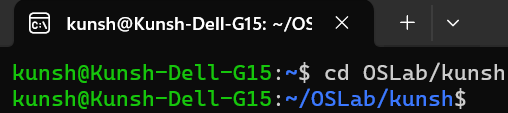
(e) List only files starting with the alphabet “N”:

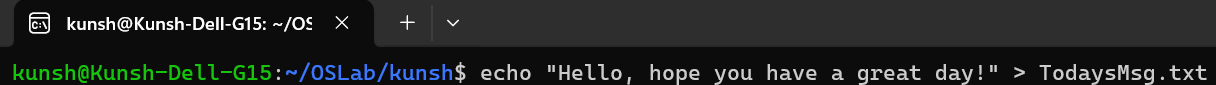
(f) Using help command, find the help on ‘man’ command:

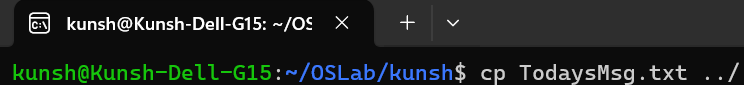
(g) Display the content of manual pages on ‘ls’ command:

(h) Demonstrate the usage of “whatis” command:

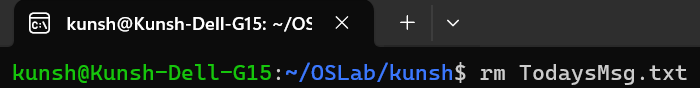
(i) Make directory named “OSLab/kunsh”.

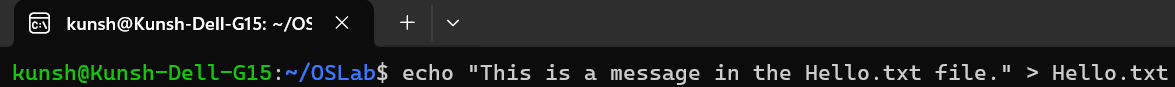
(j) Write command to reach to “kunsh” Directory:

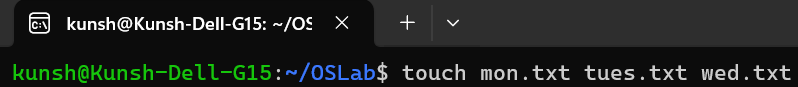
(k) Create a txt file named “TodaysMsg.txt” and write a greeting message in it:

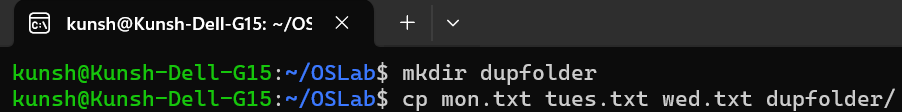
(l) Copy this file “TodaysMsg.txt” to OSLab directory:

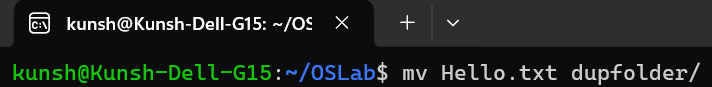
(m) Delete the file “TodaysMsg.txt” from the ‘kunsh’ Folder:

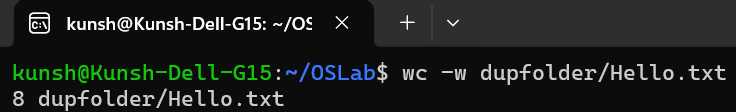
(n) Delete the directory ‘kunsh’:

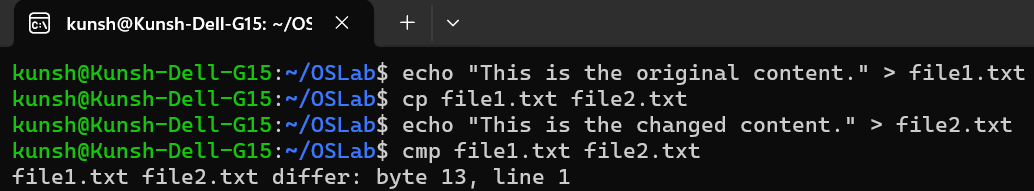
(o) Create a text file named “Hello.txt” and write a suitable message in it:

(p) Using touch command create files with names mon.txt, tues.txt, and wed.txt:

(q) Copy these newly created files to a folder named “dupfolder” after creating it:

(r) Move Hello.txt to dupfolder:

(s) Count the number of words in the Hello.txt file:

(t) Create two files with identical content, change one alphabet in one of these and compare them using cmp command:

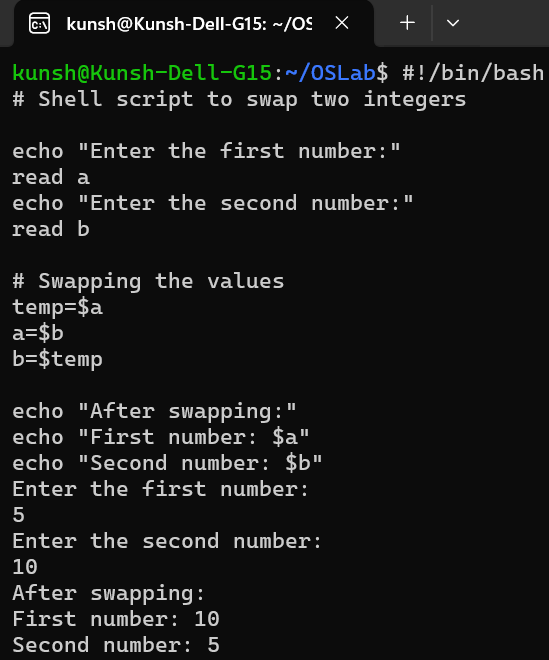
**Learning Outcome:**

**EXPERIMENT 3**

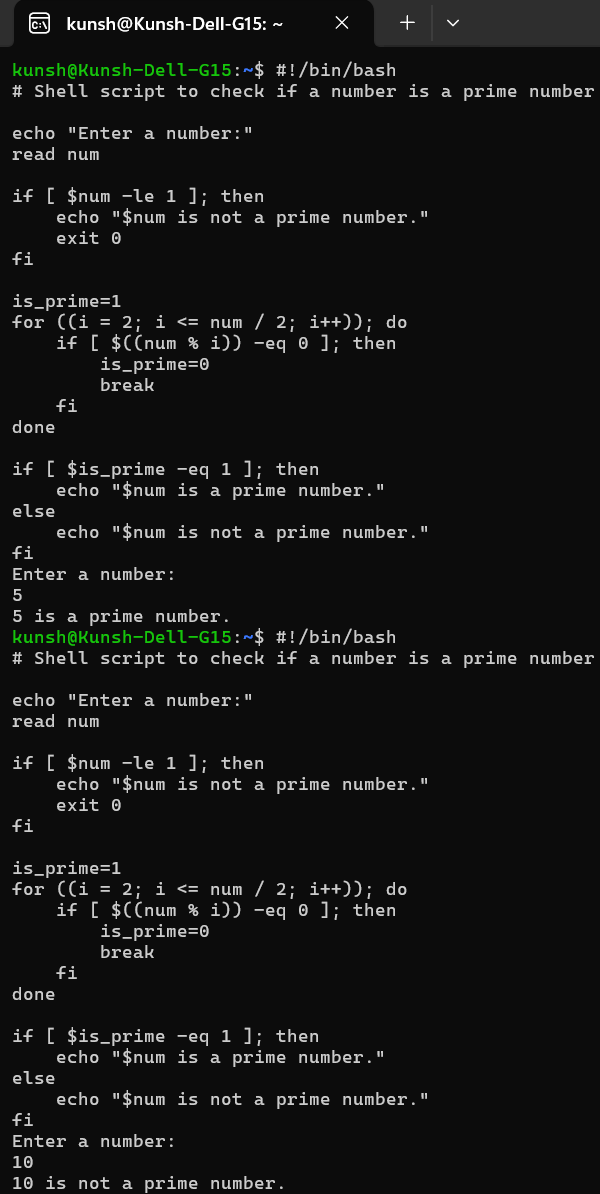
**Problem statement:** Perform following shell script-based programs:

* 1. Write a Shell Program to swap the two integers.
  2. Create a shell script that checks if a specific directory exists. If it does, the script should back up all files from that directory into a specified backup directory. The script should then loop through the files in the backup directory and list all files that were successfully copied. If the directory does not exist, the script should print an error message.
  3. Write a shell script to check if a given number is a prime number or not
  4. Write a shell script to greet the user as per the time whenever he/ she opens terminal.

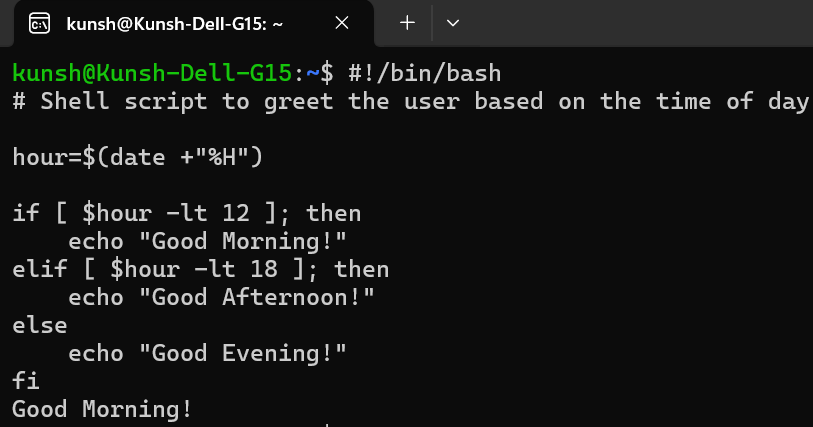
**Theory:**

**Outputs:** (a) Shell Program to Swap Two Integers:

(b) Shell Script to Back Up Files from a Directory:

(c) Shell Script to Check if a Number is Prime:

(d) Shell Script to Greet the User Based on the Time of Day:

**Learning Outcome:**

**EXPERIMENT 4**

**Problem statement:** Write a c program to implement the following scheduling algorithms.

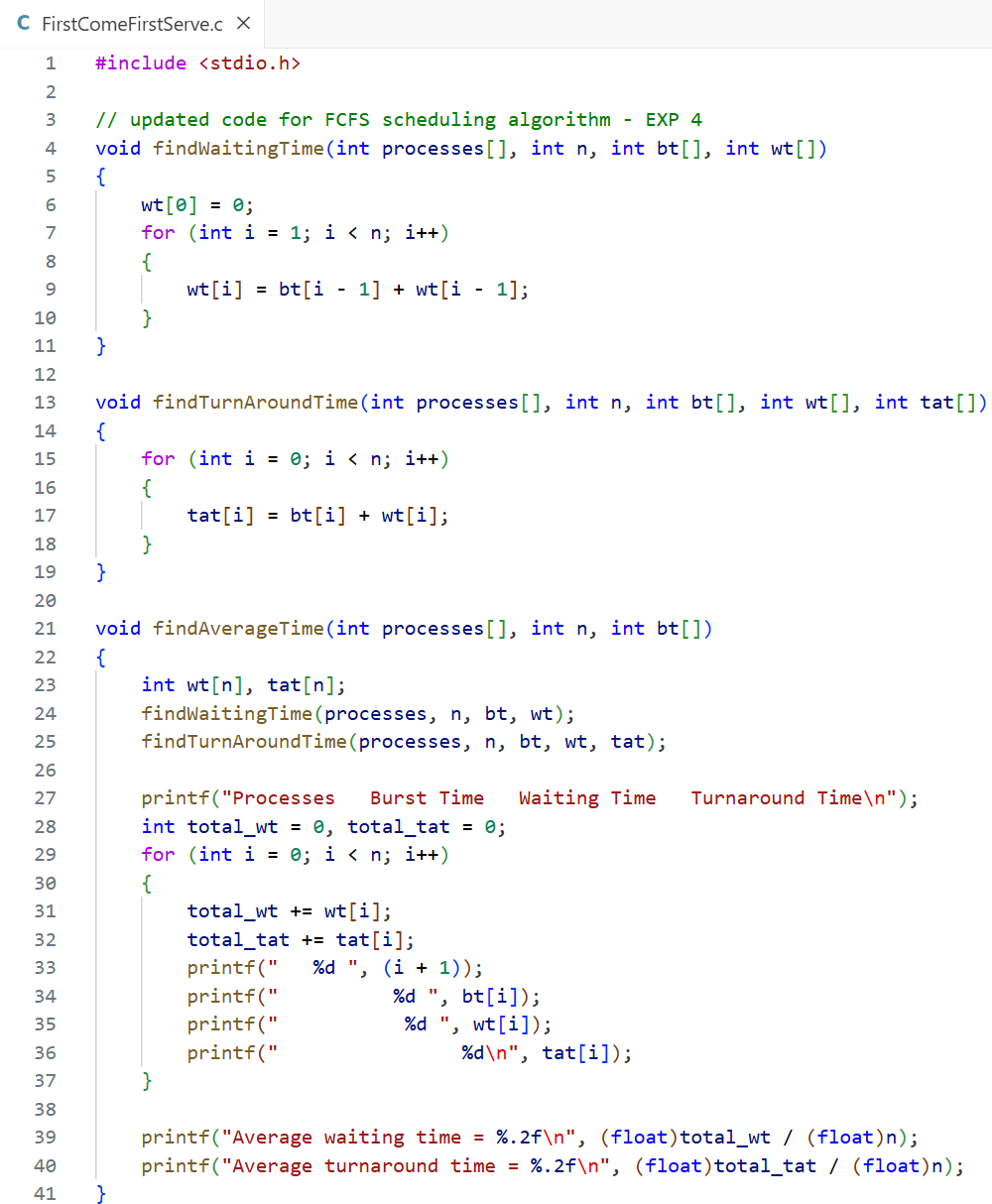
(a) First come first serve

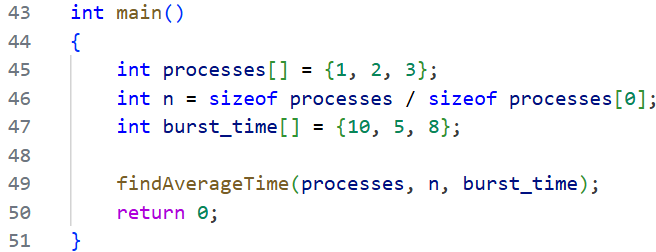
(b) Round Robin Scheduling

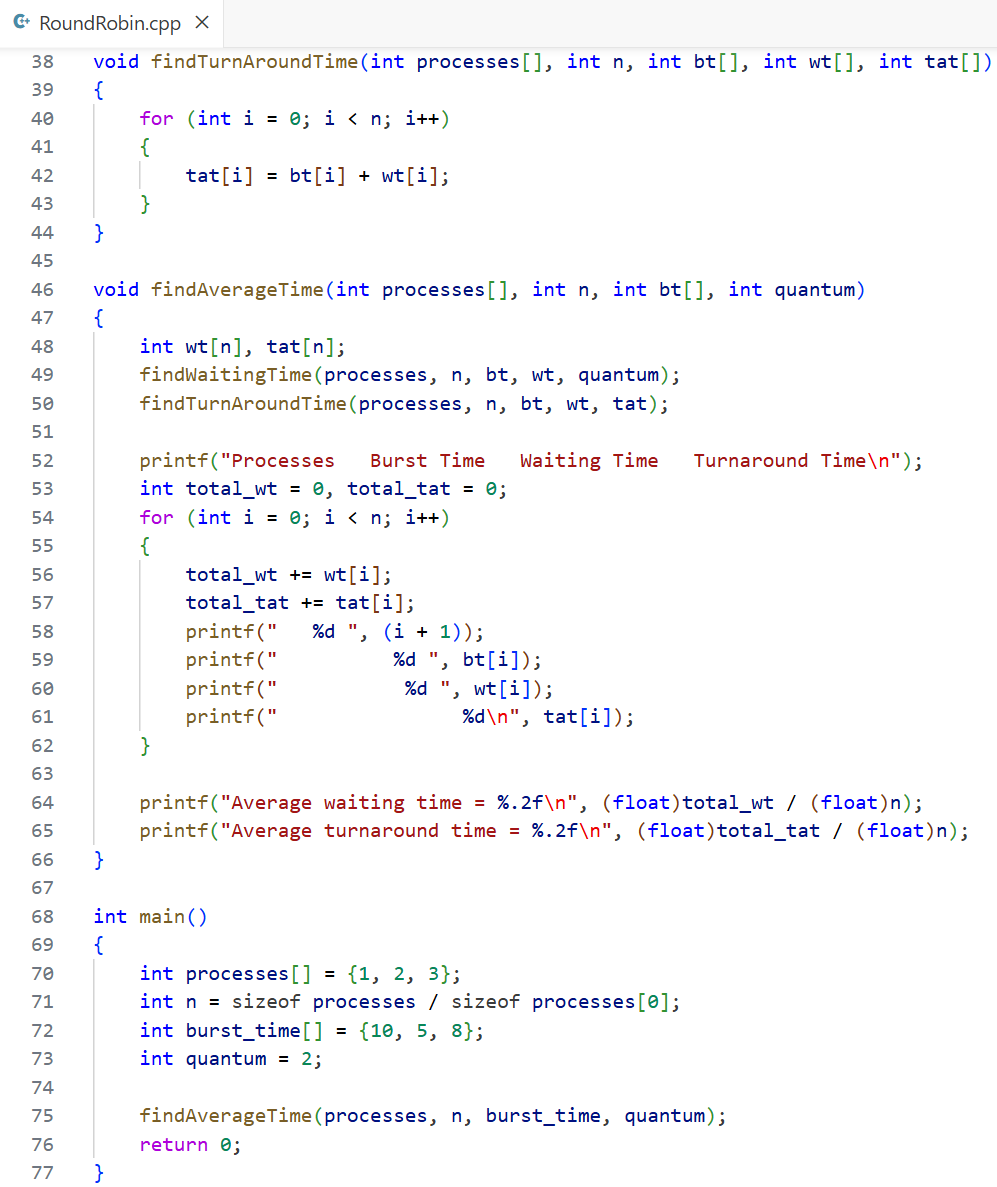
(c) Shortest job first

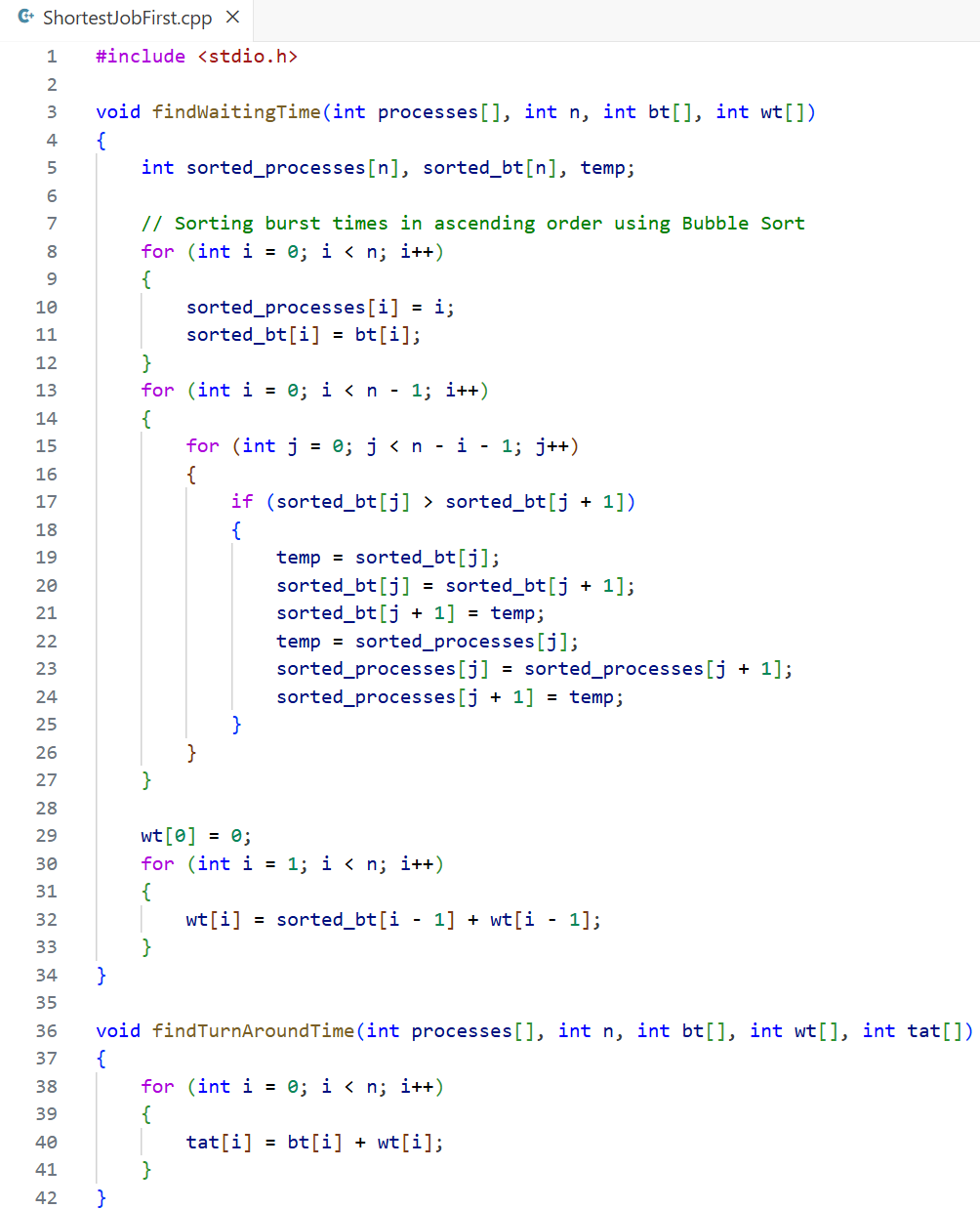
(d) Shortest Job remaining first.

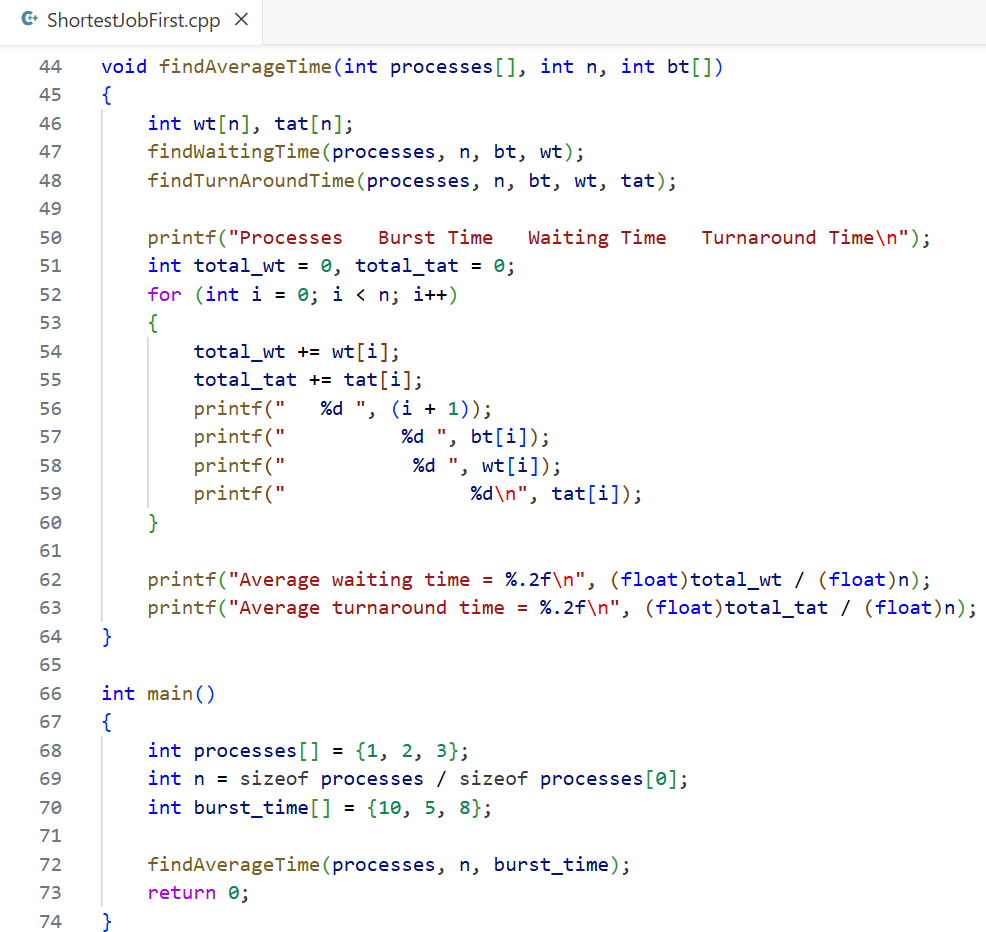
**Theory:**

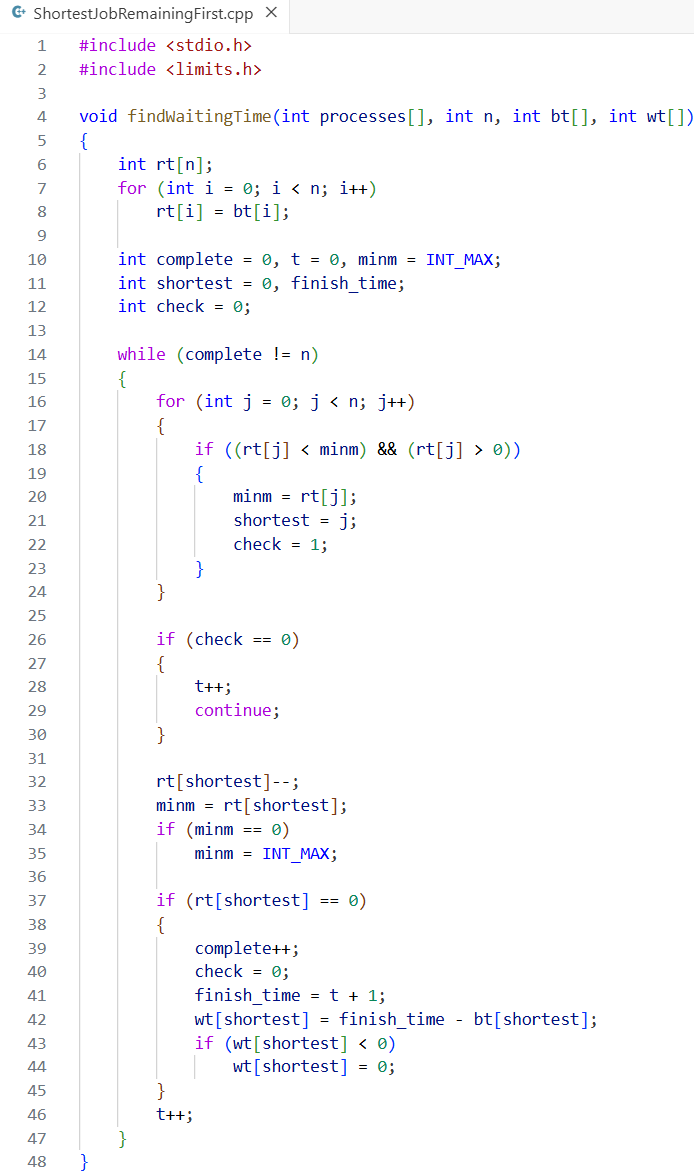
**Source Code:** (a) First Come First Serve Scheduling Algorithm

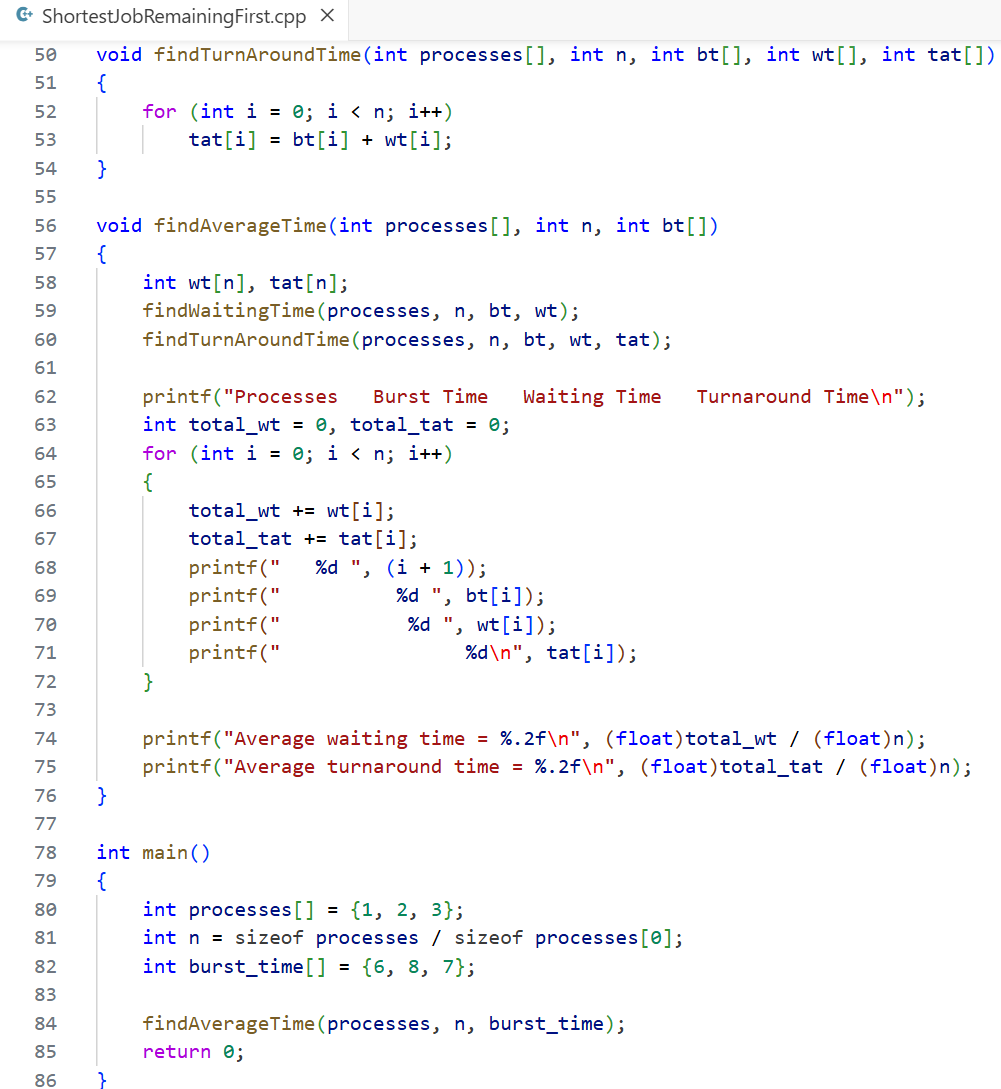
(b) Round Robin Scheduling Algorithm



(c) Shortest Job First Scheduling Algorithm

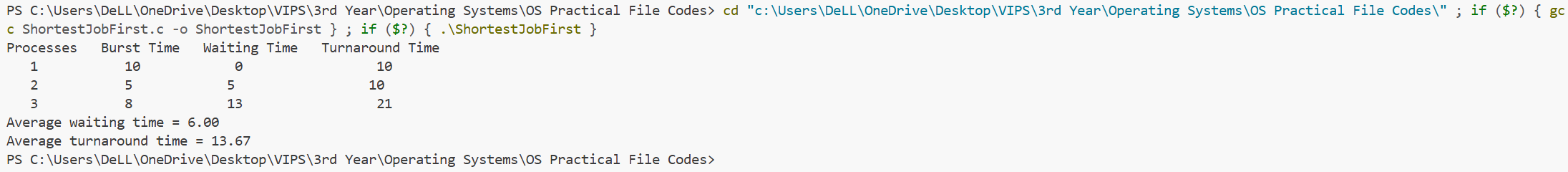


(d) Shortest Job Remaining First Scheduling Algorithm



**Outputs:** (a) First Come First Serve Scheduling Algorithm

(b) Round Robin Scheduling Algorithm

**(**c) Shortest Job First Scheduling Algorithm

**(**d) Shortest Job Remaining First Scheduling Algorithm

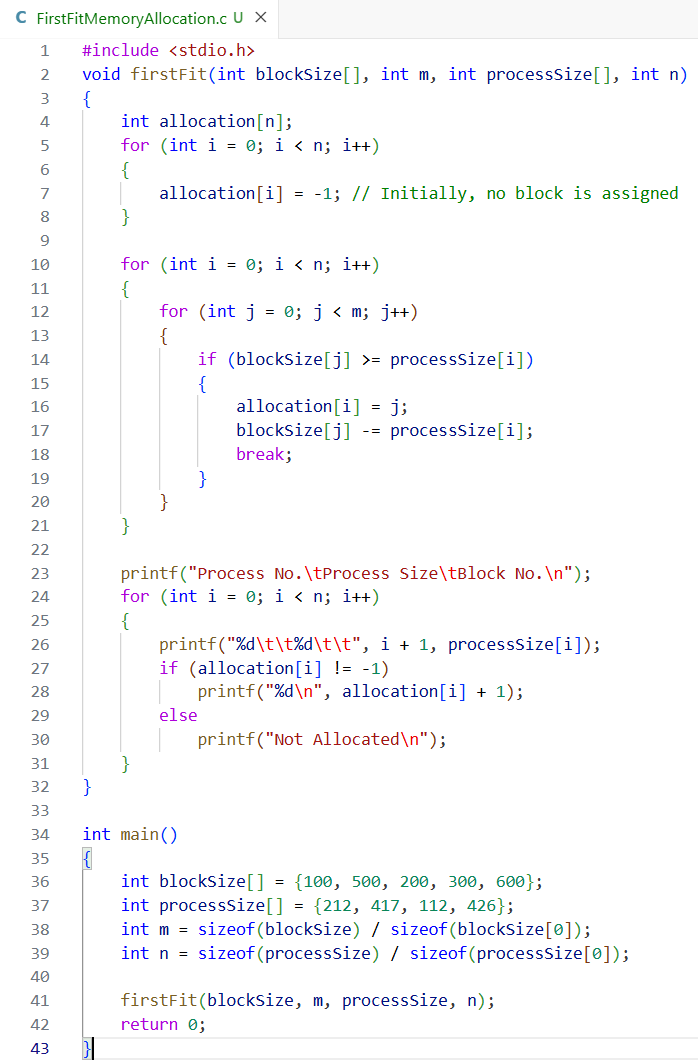
**Learning Outcome:**

**EXPERIMENT 5**

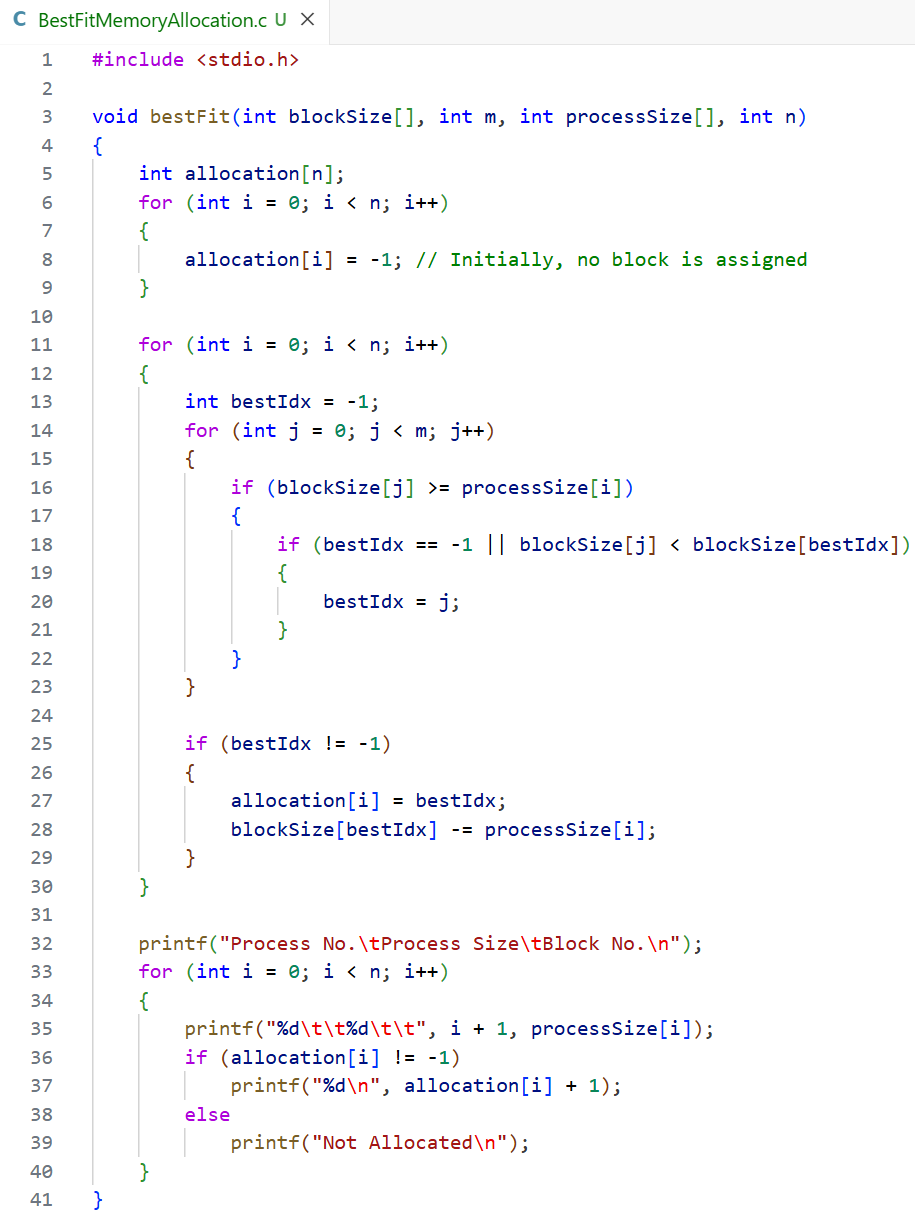
**Problem statement:** Implementation of the following Memory Allocation Methods for fixed partition

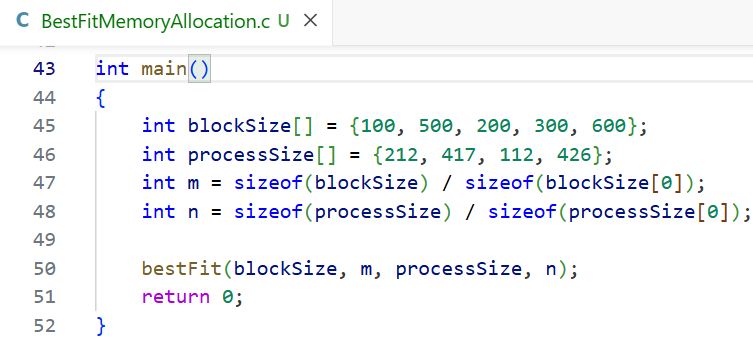
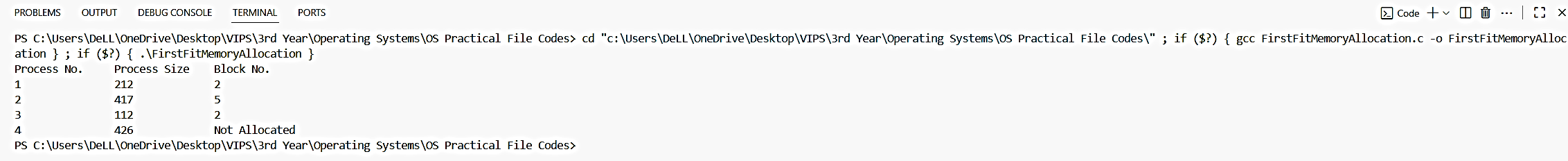
* 1. First Fit
  2. Worst Fit
  3. Best Fit.

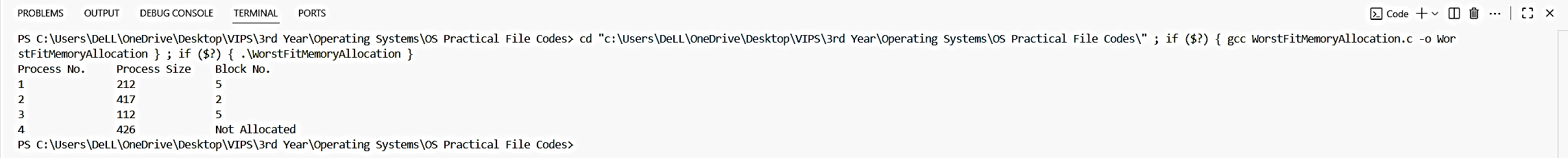
**Theory:**

**Source Code:** (a) First Fit Memory Allocation

(b) Worst Fit Memory Allocation

(c) Best Fit Memory Allocation

**Outputs:** (a) First Fit Memory Allocation

(b) Worst Fit Memory Allocation

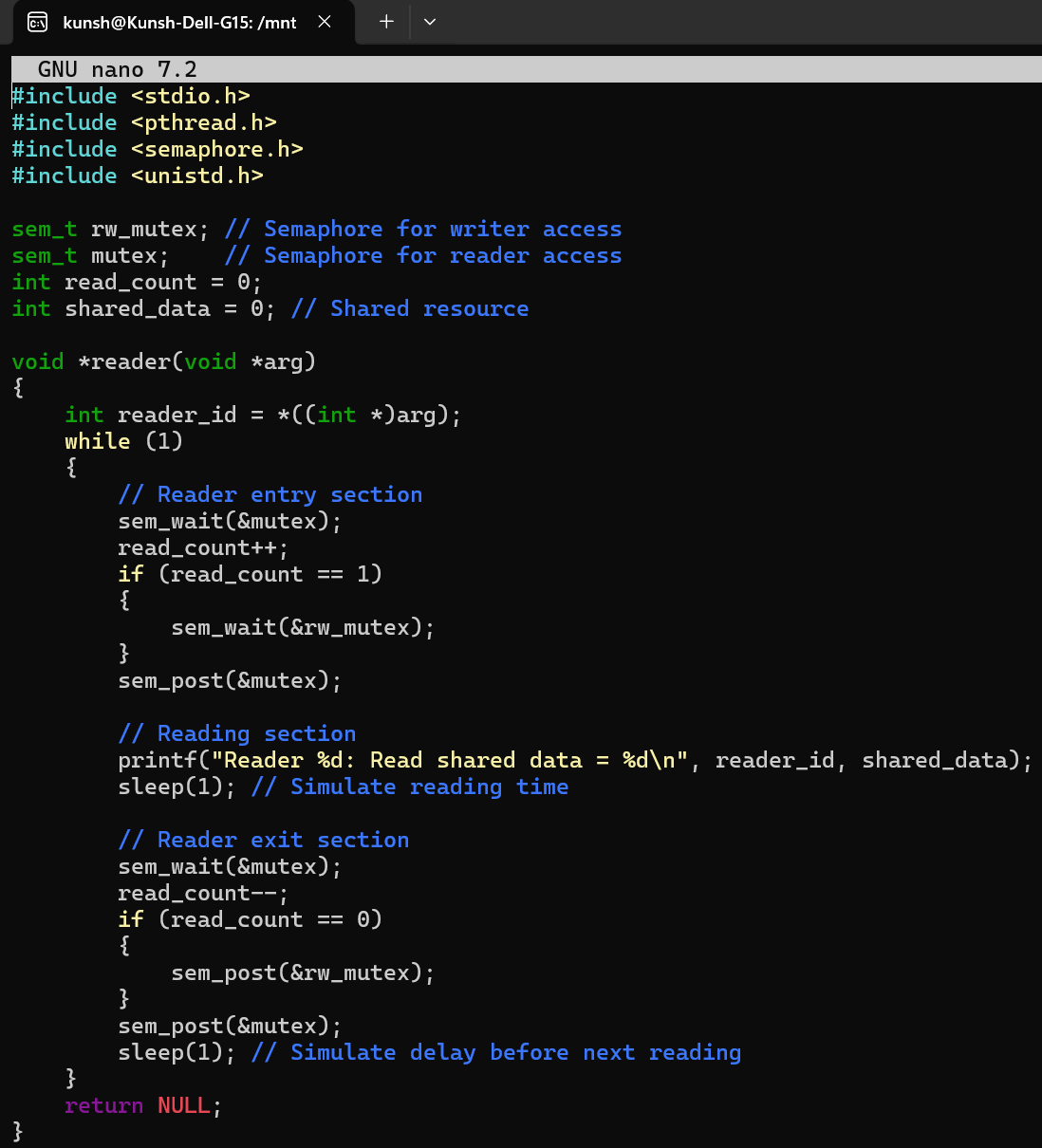
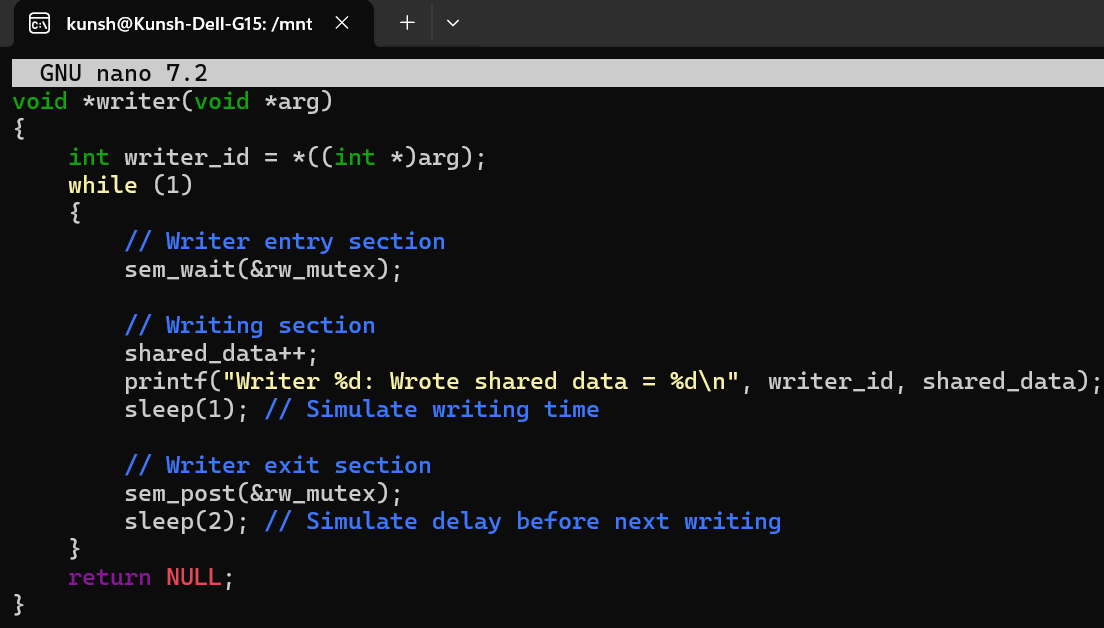
(c) Best Fit Memory Allocation

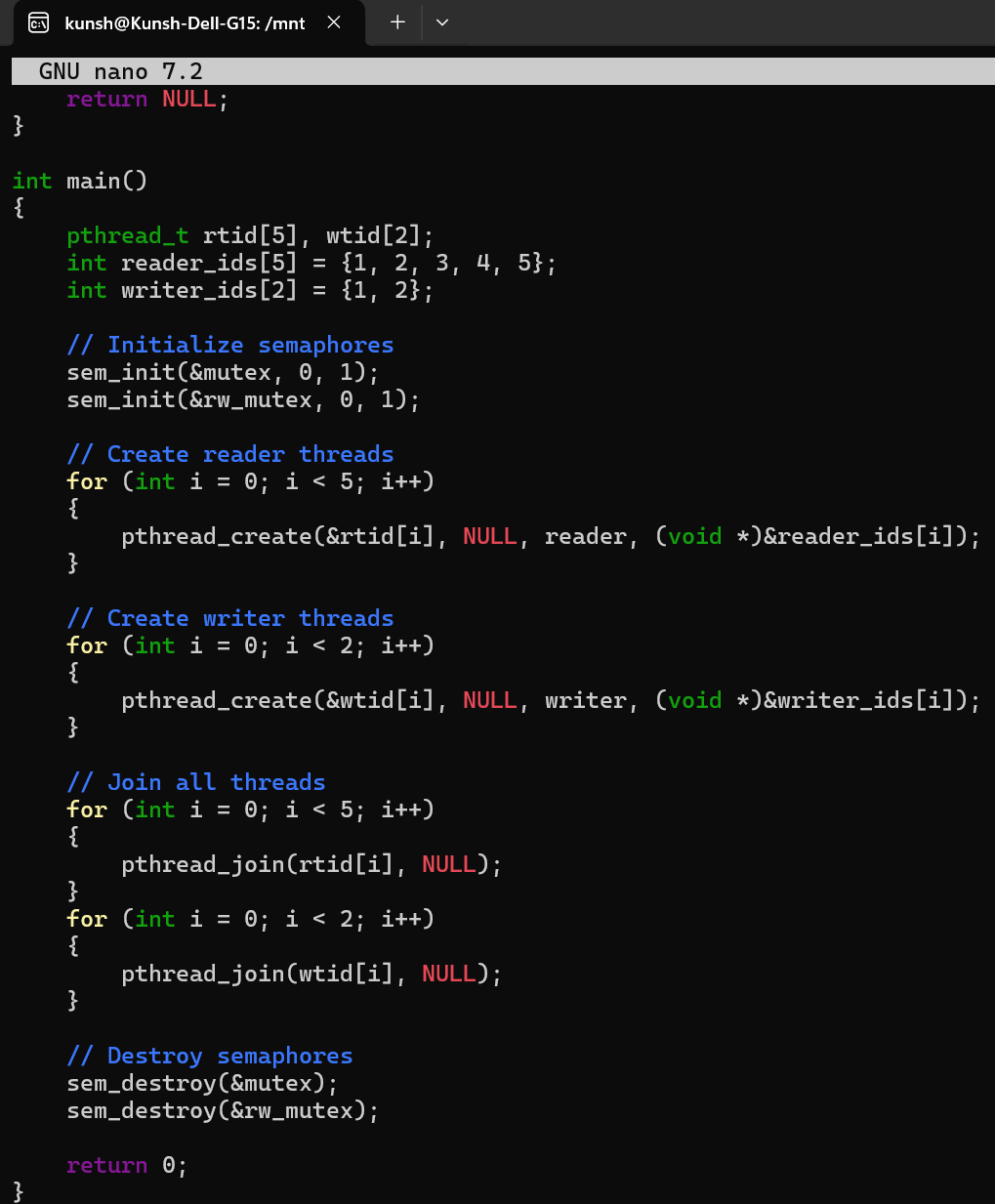
**Learning Outcome:**

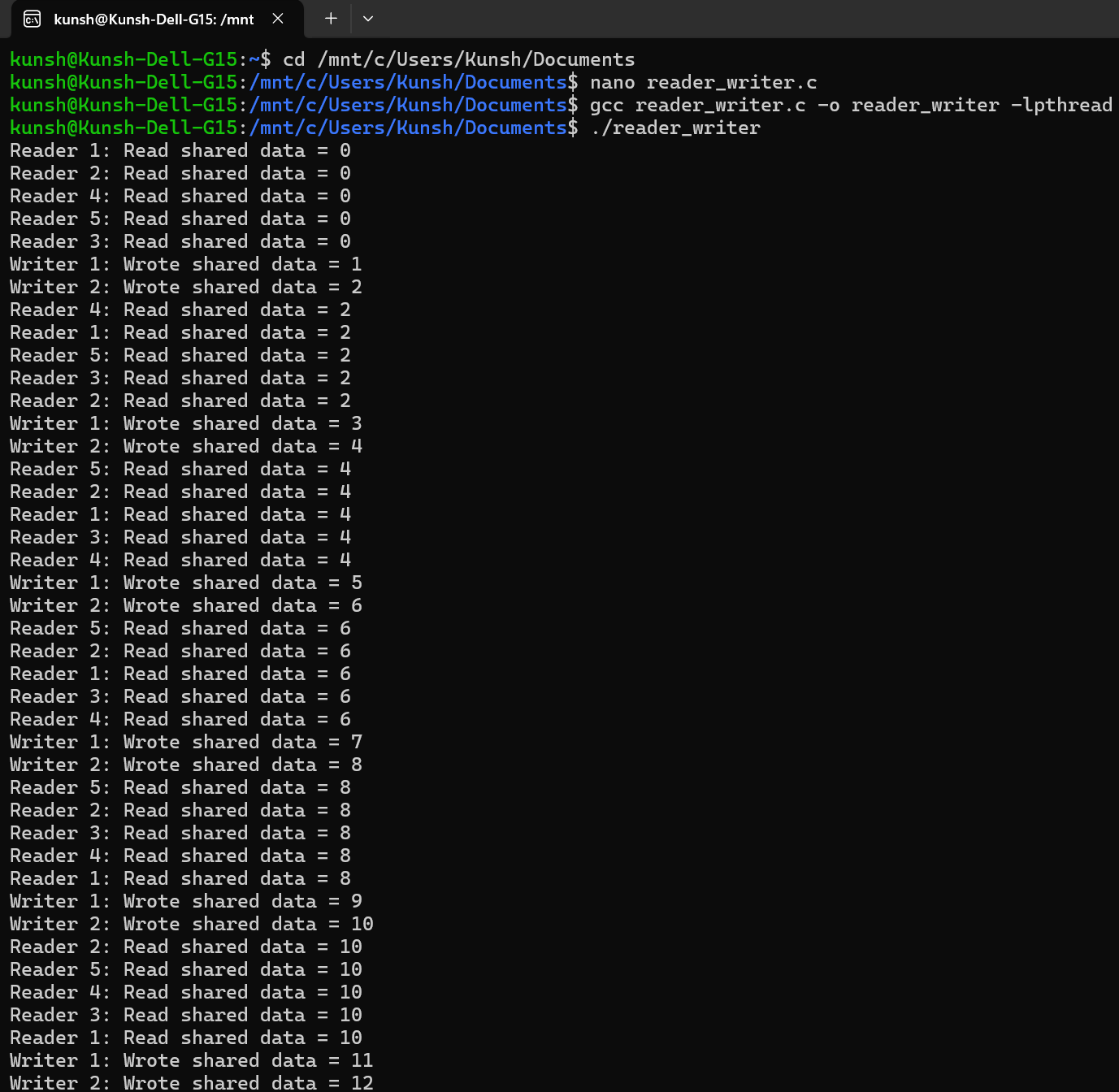
**EXPERIMENT 6**

**Problem statement:** Write a program to implement reader/writer problems using semaphore.

**Theory:**

**Source Code:**

****

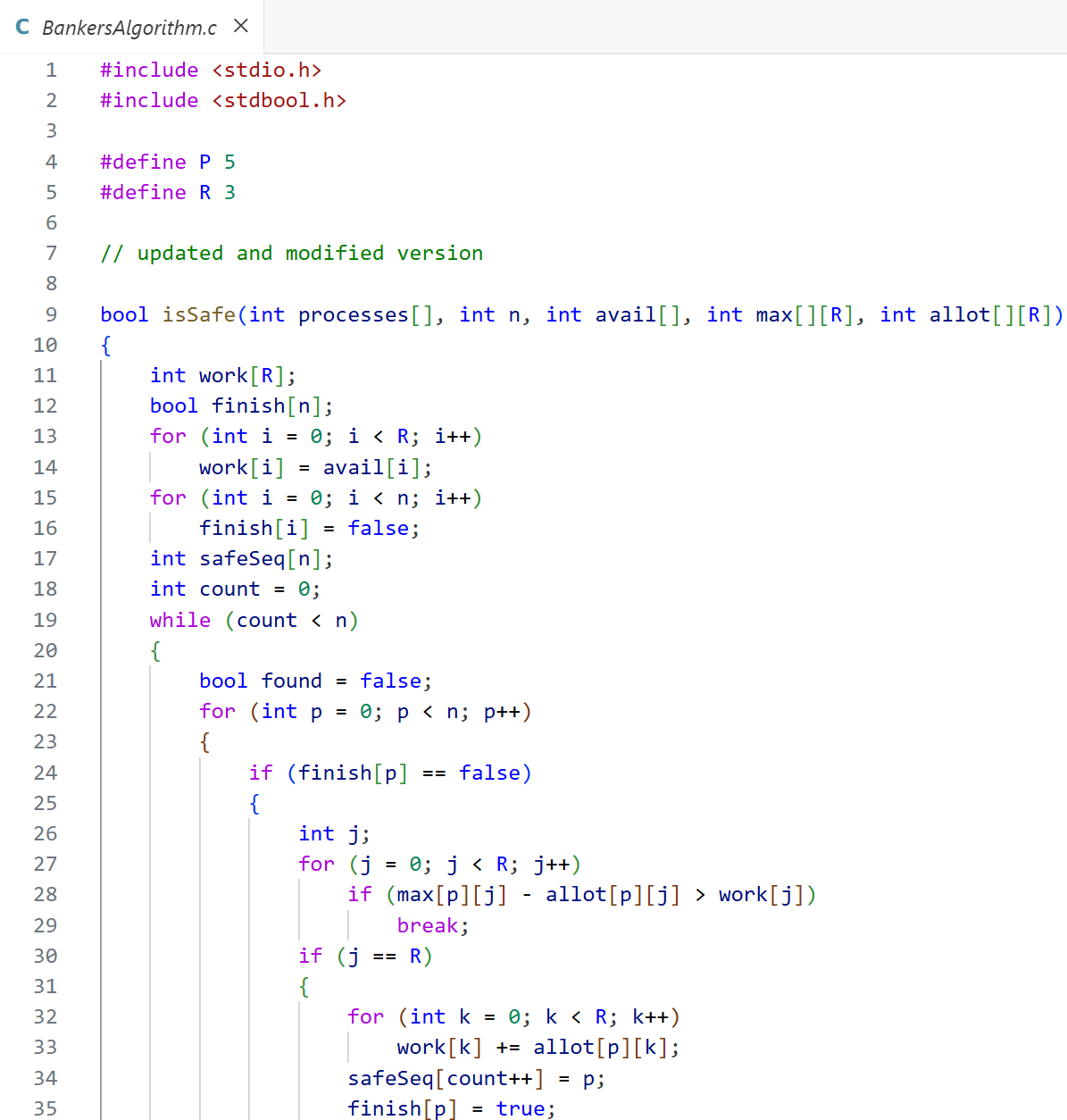
**Output:**

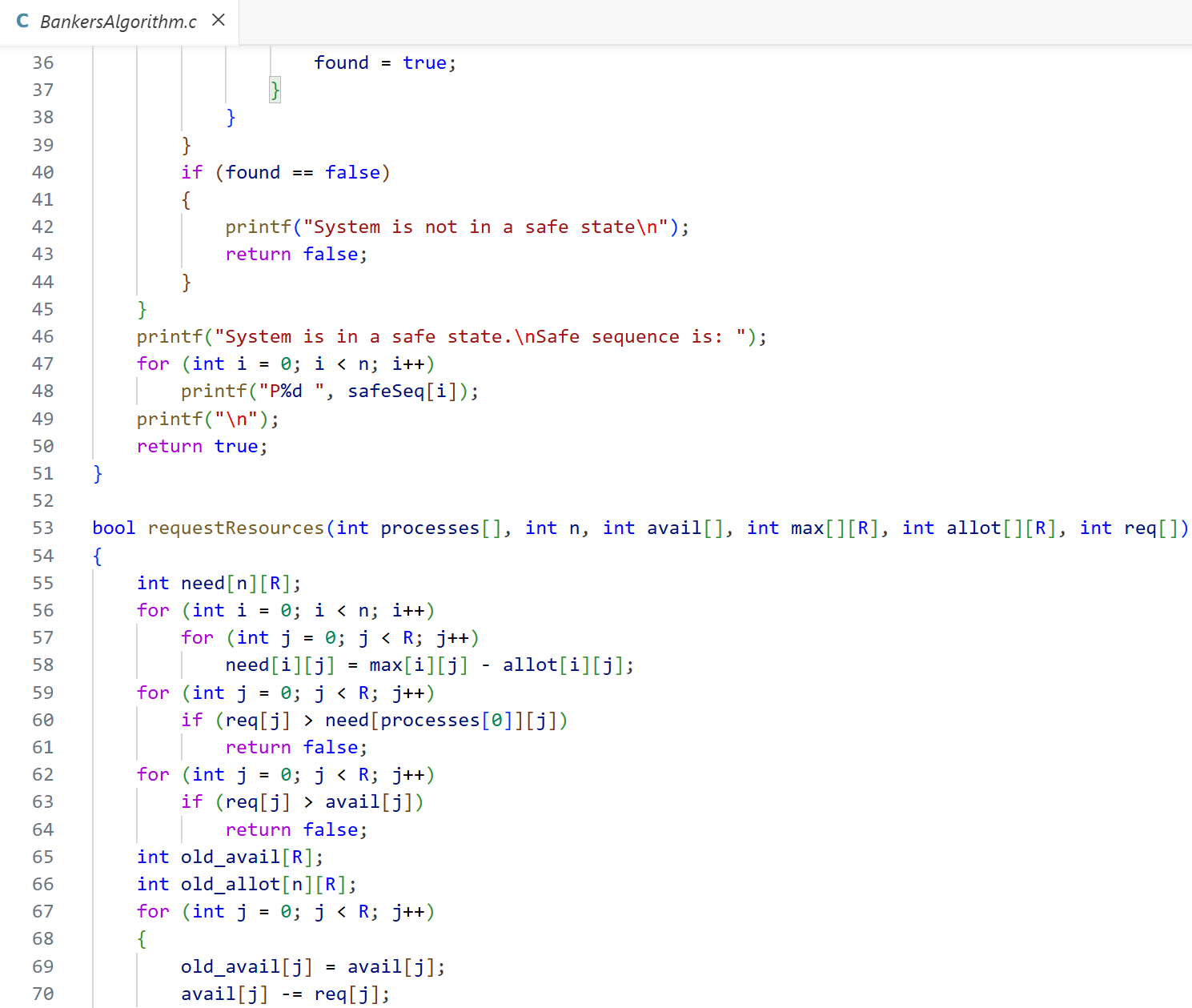
**Learning Outcome:**

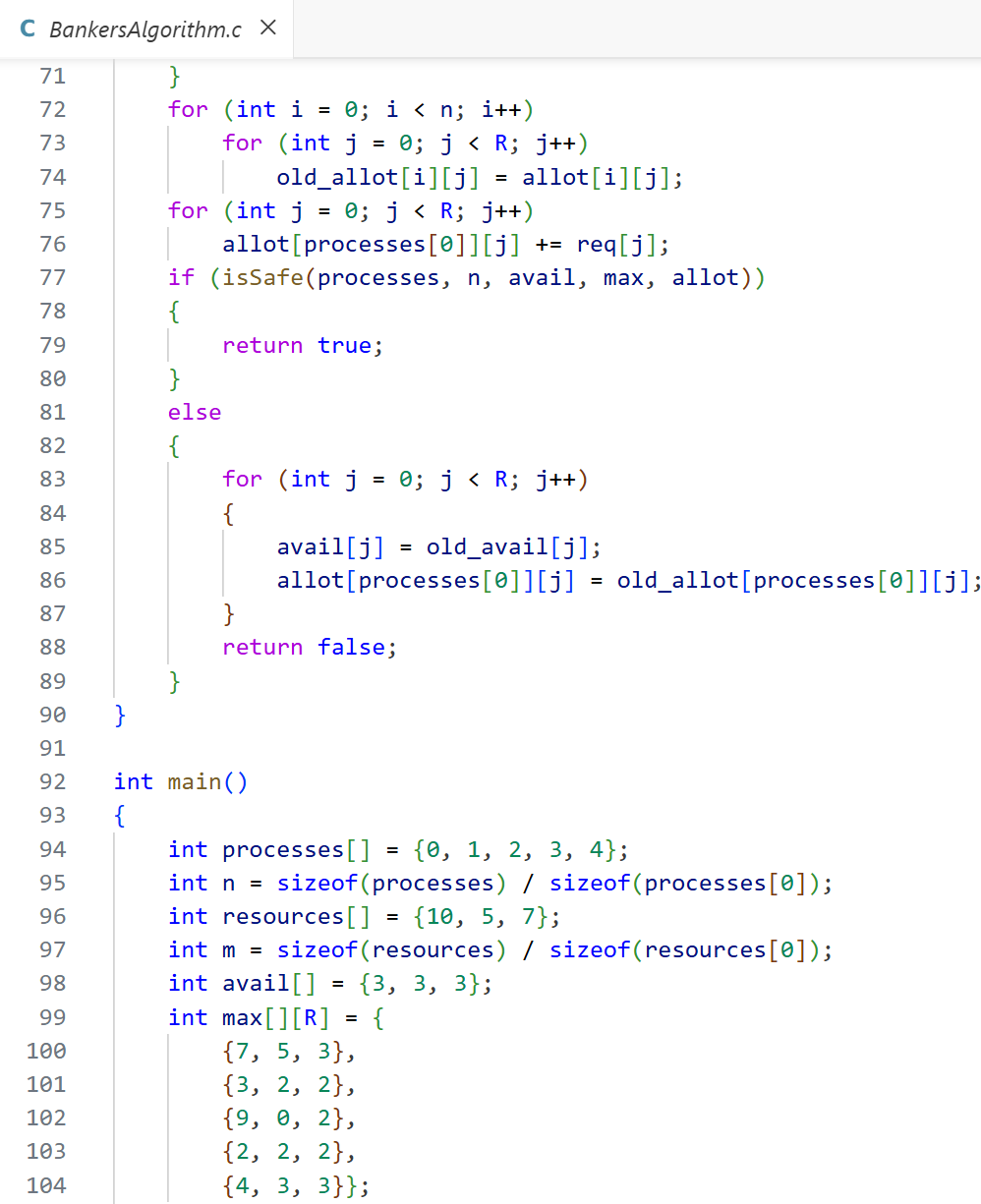
**EXPERIMENT 7**

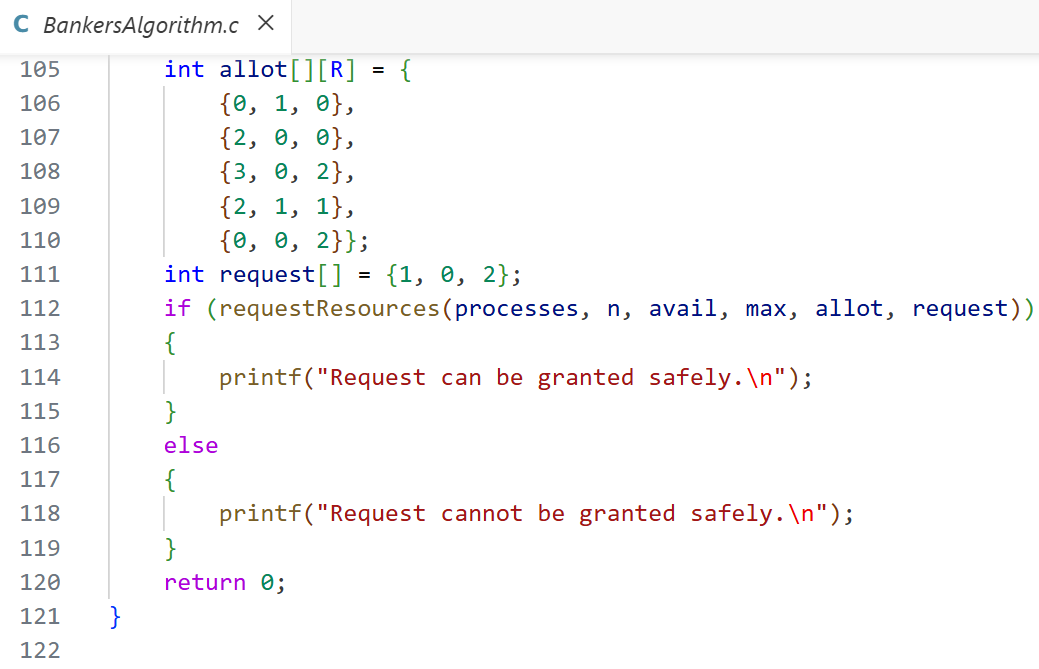
**Problem statement:** Write a program to implement Banker’s algorithm for deadlock avoidance.

**Theory:**

**Source Code:**

****

****

**Output:**

**Learning Outcome:**

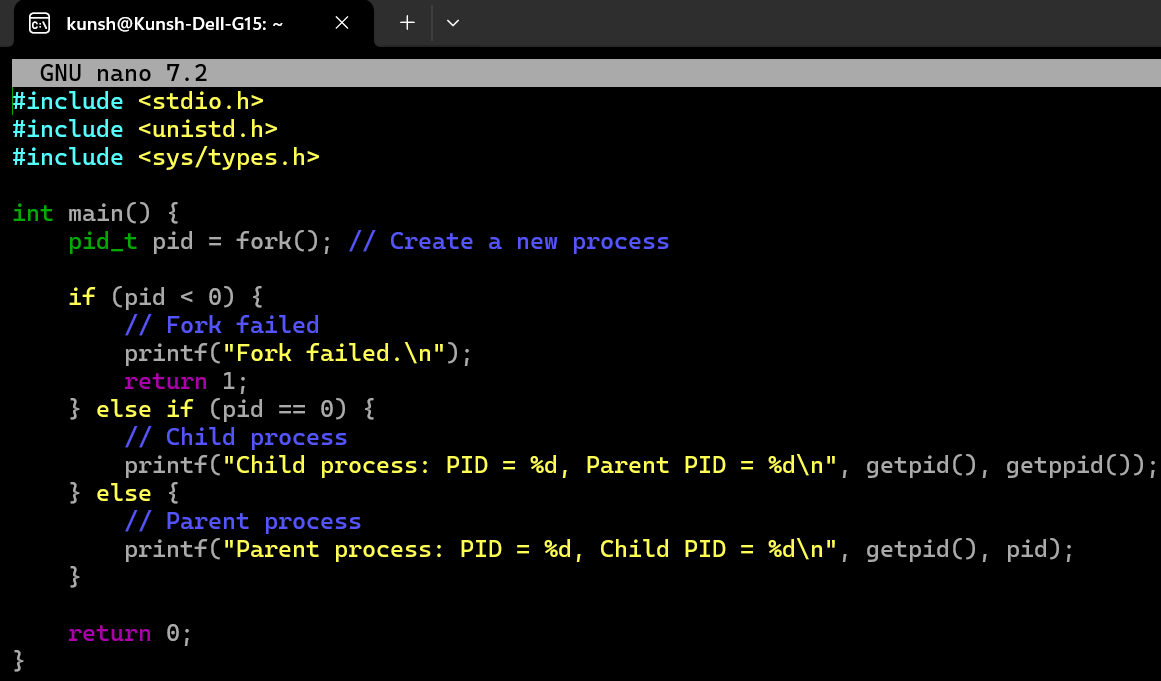
**EXPERIMENT 8**

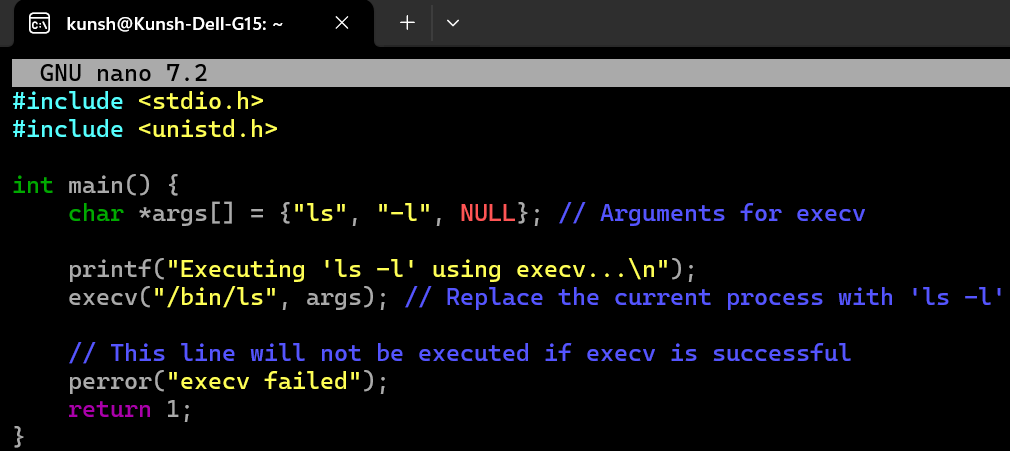
**Problem statement:** Perform following related to process management: -

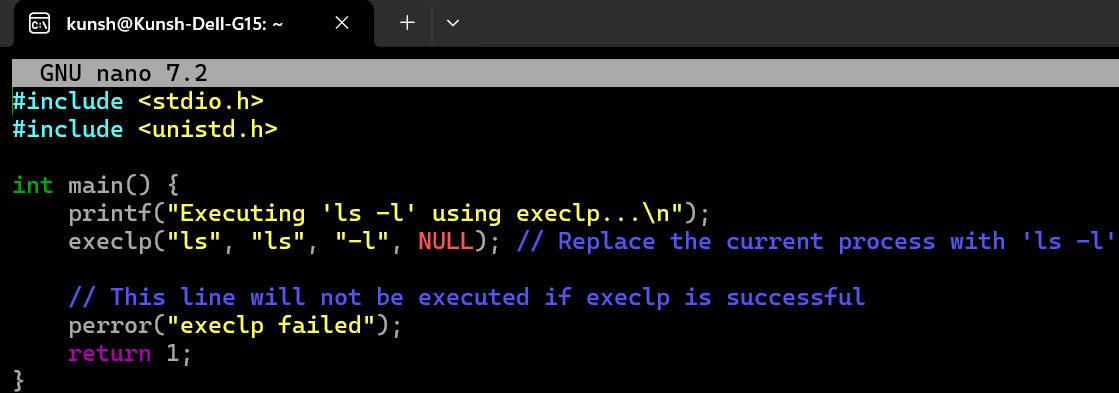
* Program to implement the fork function using C.
* Program to implement execv function using C.
* Program to implement execlp function.
* Program to implement wait function using C.
* Program to implement sleep function using C.

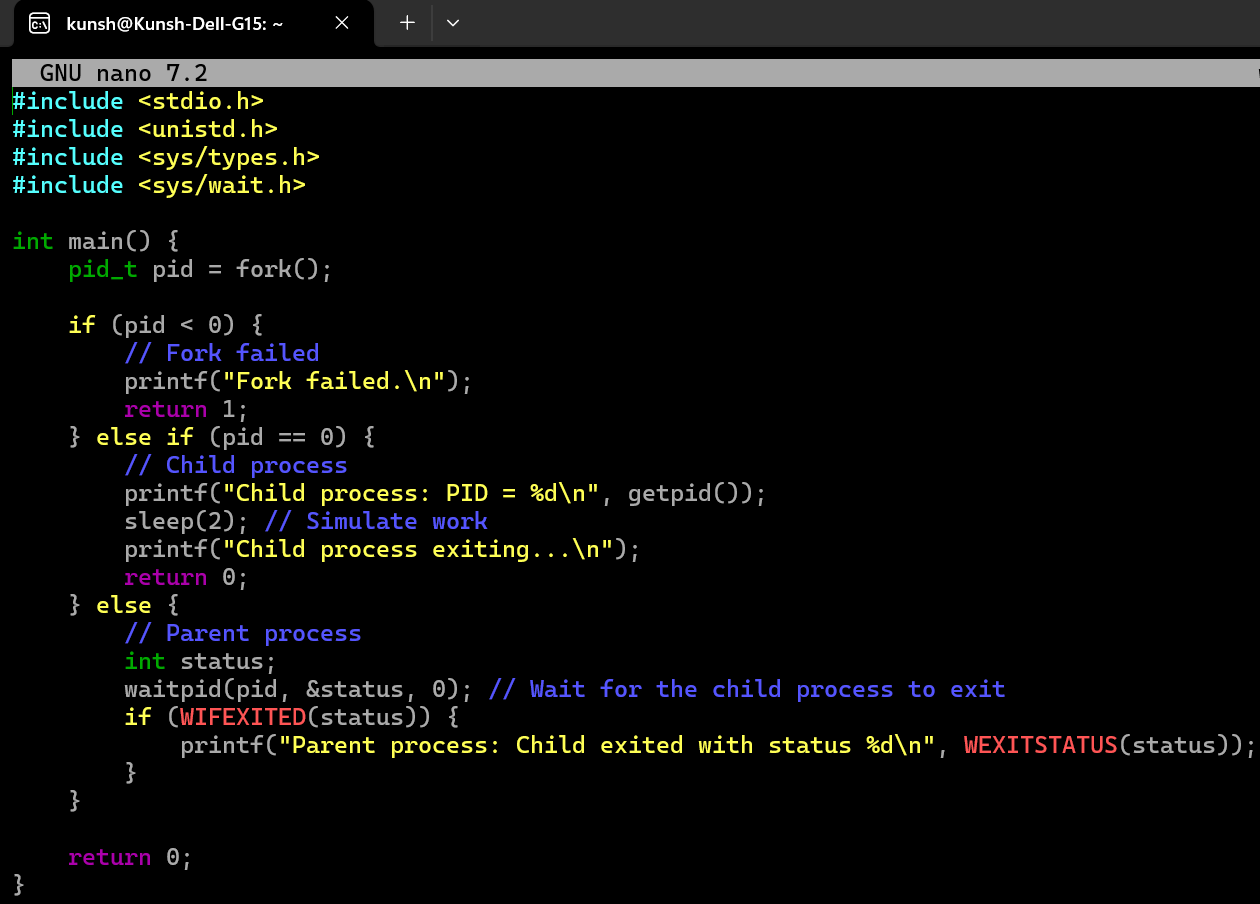
**Theory:**

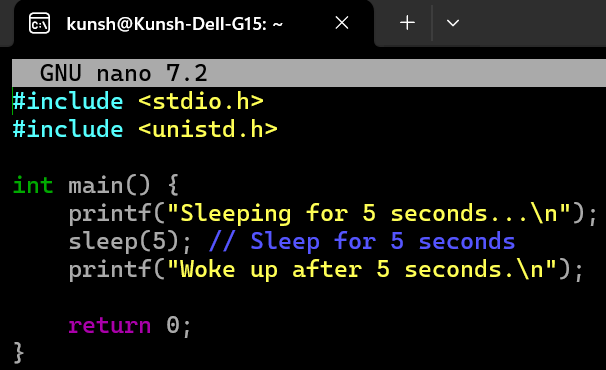
**Source Code:**

(a) Program to implement the fork function using C:

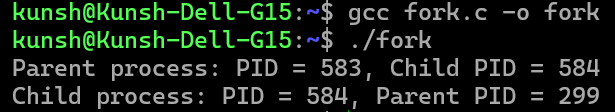
(b) Program to implement execv function using C:

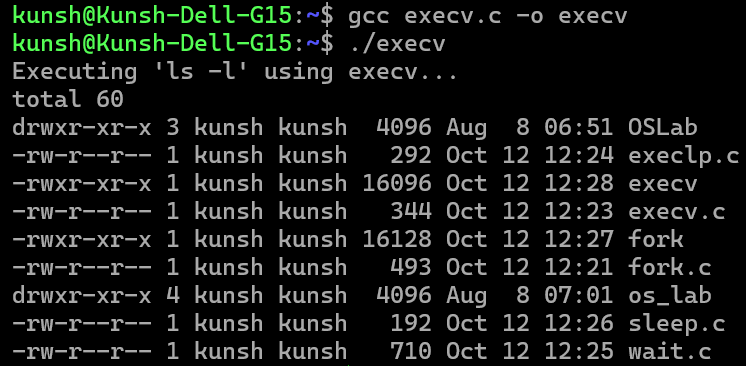
(c) Program to implement execlp function:

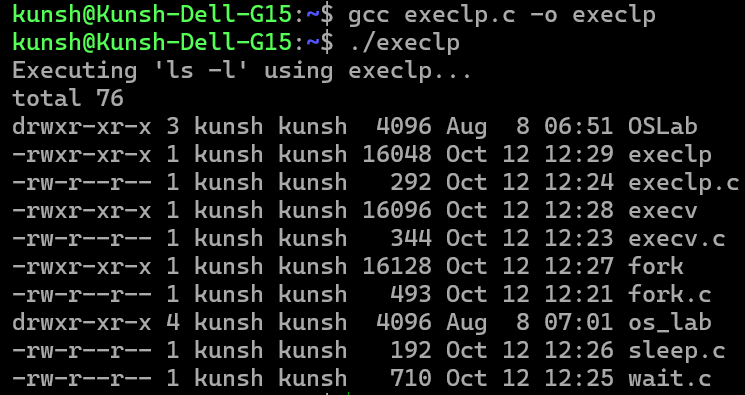
(d) Program to implement wait function using C:

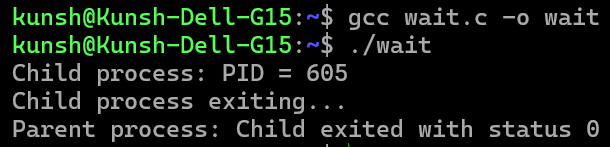
(e) Program to implement sleep function using C:

**Output:**

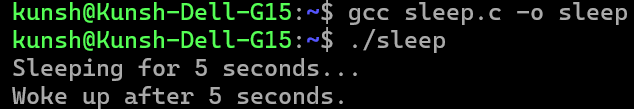
(a) Program to implement the fork function using C:

(b) Program to implement execv function using C:

(c) Program to implement execlp function:

(d) Program to implement wait function using C:

(e) Program to implement sleep function using C:

**Learning Outcome:**

**EXPERIMENT 9**

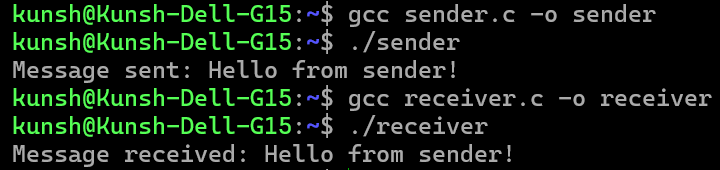
**Problem statement:** Write a program to implement Inter Process Communication (IPC) using Message Queues.

**Theory:**

**Source Code:**

(a) Sender End Code:

(b) Receiver End Code:

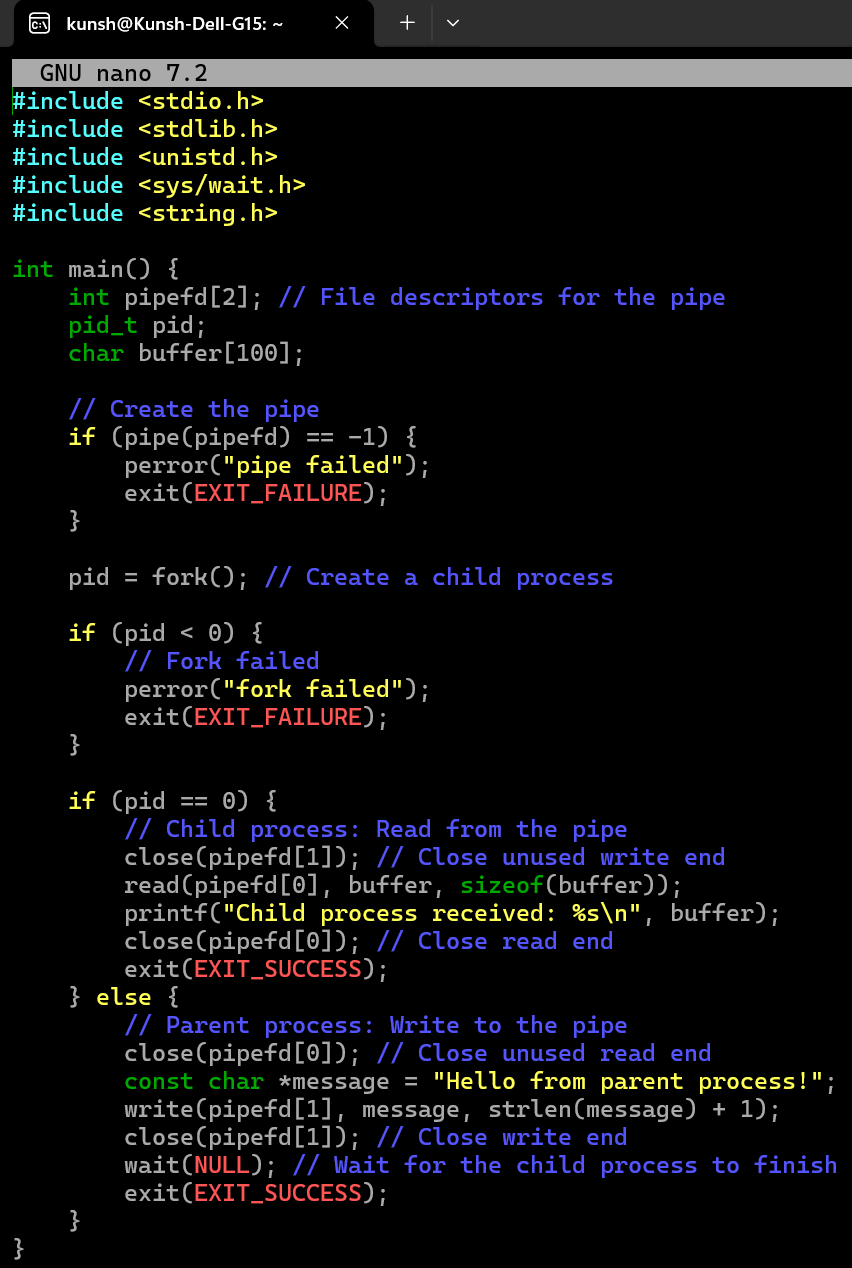
**Output:**

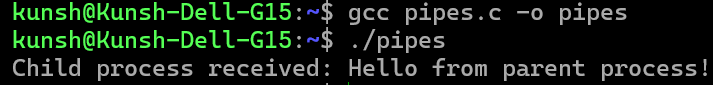
**Learning Outcome:**

**EXPERIMENT 10**

**Problem statement:** Write a program to implement IPC using pipes.

**Theory:**

**Source Code:**

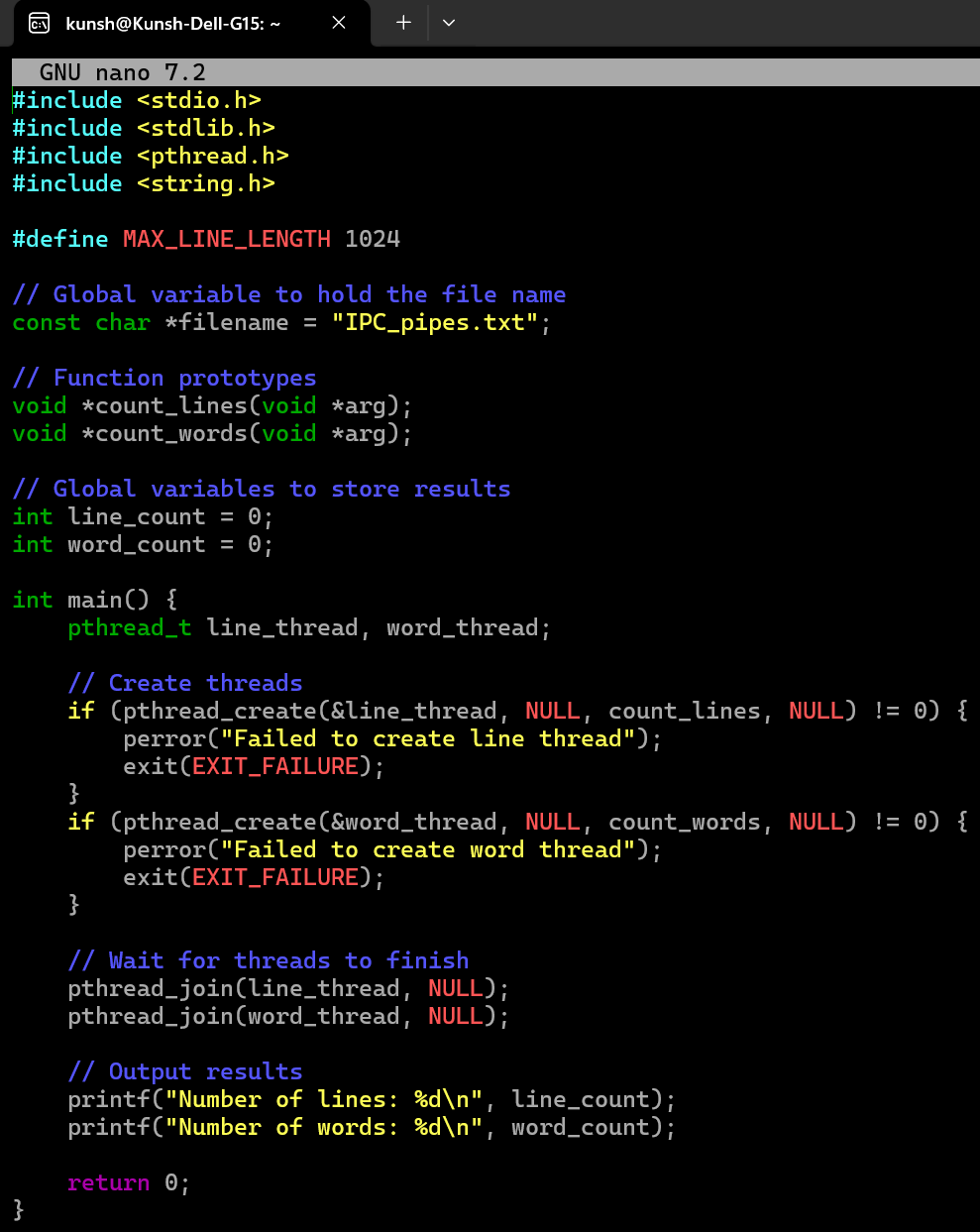
**Output:**

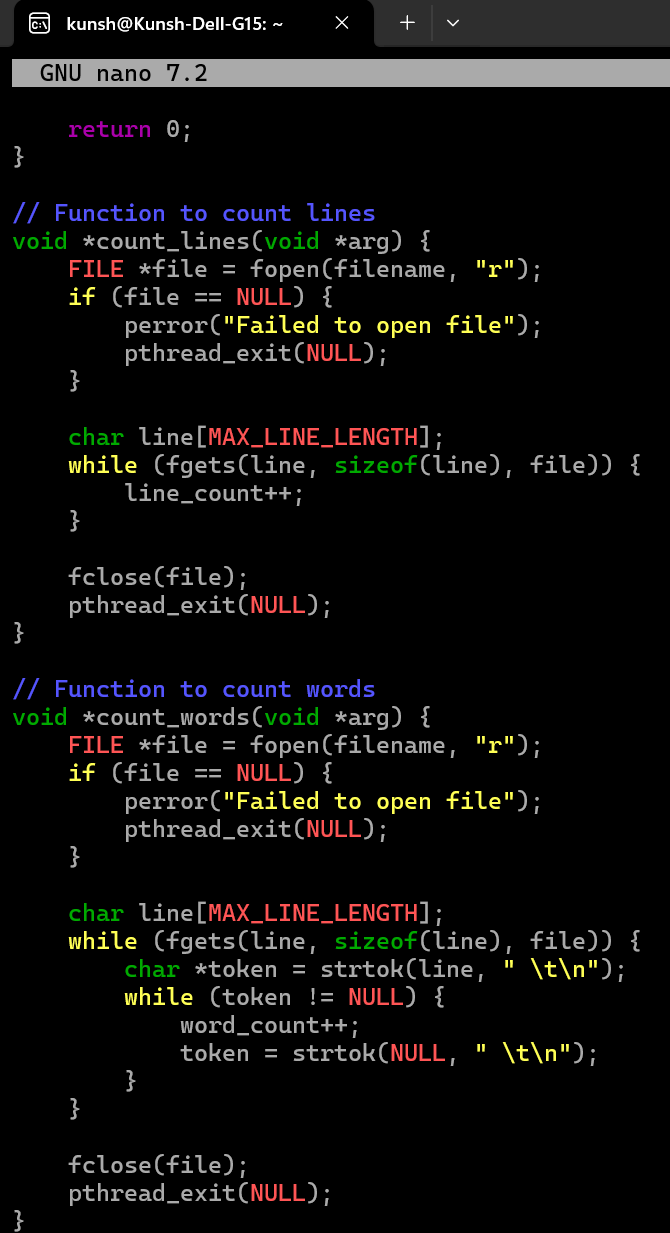
**Learning Outcome:**

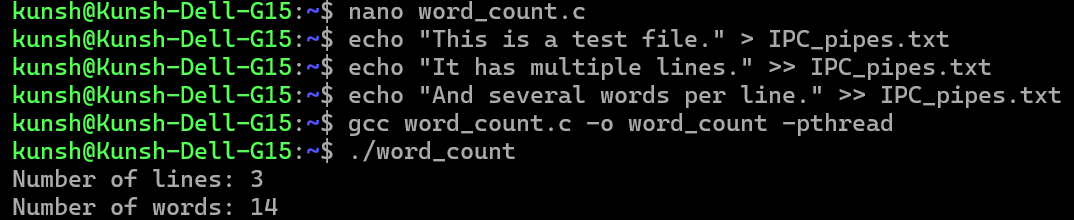
**EXPERIMENT 11**

**Problem statement:** Write a program using Pthread, where main thread calculates number of lines in a file and child calculates number of words.

**Theory:**

**Source Code:**

****

**Output:**

**Learning Outcome:**