**Assignment Objective**

The goal is to create a user-defined process, find its Process Identification Number (PID), and verify this PID using the top command. This exercise helps in understanding process management in Linux.

**Detailed Steps**

**1. Write a C Program to Display PID**

Create a C program that will display its PID and run indefinitely. This ensures that the process will stay active long enough for you to check it with the top command.

**Code Explanation:**

#include <stdio.h>

#include <unistd.h>

int main() {

// Print the PID of the current process

printf("Process ID: %d\n", getpid());

// Infinite loop to keep the process running

while (1) {

// The process will keep running

}

return 0;

* }#include <stdio.h> and #include <unistd.h>: Includes the necessary headers for I/O operations and to access system calls.
* getpid(): A system call that returns the PID of the calling process.
* while (1): An infinite loop that keeps the process running.

**2. Compile the Program**

Use the gcc compiler to compile the C program.

gcc display\_pid.c -o display\_pid

* display\_pid.c: The source file name.
* -o display\_pid: Specifies the name of the output executable file.

**3. Run the Program**

Execute the compiled program in a terminal.

./display\_pid

This will print the PID and then enter an infinite loop, keeping the process active.

**4. Use the top Command**

Open a second terminal window to monitor system processes.

Top : The top command provides a real-time view of the system's processes, including their PIDs, CPU usage, memory usage, etc.

**5. Locate Your Process in top**

In the top output, locate your process by its PID:

1. **Sort by PID**:
   * Press Shift + O (uppercase 'o') to sort by different columns.
   * Select PID if it’s not already sorted by PID.
2. **Filter by Process Name (optional)**:
   * Press Shift + F to choose columns and add COMMAND if it's not displayed.
3. **Find the PID**:
   * Look for the PID that matches the one printed by your program.

**6. Verify the PID**

Check if the PID displayed by your program matches the PID shown in top. This verifies that the process you created is indeed the one being monitored.

**Summary of Assignment**

1. **Write a Program**: Create a C program that prints its PID and runs indefinitely.
2. **Compile the Program**: Use gcc to compile the program.
3. **Run the Program**: Execute it to print and display its PID.
4. **Monitor with top**: Use the top command in another terminal to observe processes.
5. **Verify**: Locate your process by PID in the top output to ensure it's the same as printed by the program.

**Learning Outcome**

This assignment helps you understand:

* How to create and manage processes in Linux.
* How to retrieve and verify the PID of processes.
* How real-time process monitoring works using tools like top.

Feel free to ask if you need further clarification or assistance with any part of this assignment!

4o mini