Assignment 2

Name:Mohamed hatem ahmed said

ID: 23101591

TASK 1 :Write the code examples you have seen in Lab-5. Explain each one
and what it does.

Exercise 1: Using fork() in C

code demonstrates how a program can create a new process using fork() allowing the parent and child processes to run independently, which is useful for multitasking, parallel execution, and process management in operating systems.

```
#include <stdio.h>
#include <unistd.h>
int main() {
pid_t pid = fork();
if (pid == 0) {
printf("This is the child process. PID: %d\n", getpid());
} else if (pid > 0) {
printf("This is the parent process. PID: %d\n", getpid());
} else {
printf("Fork failed!\n");
}
return 0;
}
Task: Compile the program
gcc process_creation.c -o process_creation
./process_creation
```

Exercise 2: Starting Processes in the Background

starts the sleep program in the background and allowing the terminal to remain usable while the process waits for 300 seconds it's used to run long tasks without blocking the shell.

The command is : **sleep 300 &**

Exercise 3: Stopping Processes

1- searches all running processes for any that contain the word "sleep"
ps aux | grep sleep
2- Stop the process using kill:
kill <PID>
3- Verify the process is no longer running:
ps aux | grep sleep

Exercise 4: Pausing and Resuming a Process

1- This command pauses the process without terminating it

kill -STOP <PID>

2- This command esumes a previously paused process and continues running it normally

kill -CONT <PID>

TASK2,3 : JOB OF LINKER /LOADER:

1-LINKER:-

> The linker combines multiple object files into a single executable, resolving symbol references between modules and connecting needed library functions. It works after compilation and produces the final executable program.

2-LOADER:-

The loader, part of the operating system, loads this executable into memory, allocates space for code and data, initializes the program's environment, and adjusts memory addresses as needed. It may also load libraries dynamically at runtime.

Source Code → COMPILER → Object Code → LINKER → Executable Files → LOADER → MEMORY

Additional Files → LINKER