

OS Practical - 4 - Processes in Linux

- (1) Aim:- Process Management & Process Creation in Linux using C Programming
- (2) Tools =>
 - Linux Operating System (Ubuntu) using fork()
 - Terminal
 - GCC compiler
 - C Programming Language
- (3) Procedure/steps:-

part 1: killing processes in Linux

- (1) First, open the Terminal in Linux.
- (2) To view all running processes, use the command:
=> `ps -ef`
- (3) To kill a process by process name, use:
`killall process_name` => `killall firefox`
- (4) To kill a process based on process name using `pkill`, use:
`pkill process_name`
- (5) To kill a single process using process ID (PID):
 - First find PID using:-
`ps -ef | grep process_name`
 - Then kill the process using:- `kill PID`
- (6) If the process does not terminate, force kill it using:- `kill -9 PID`

Process Creation using C Program

- (1) Create a C file using terminal:- `nano process.c`
- (2) Write a C program using the `fork()` system call.
- (3) Compile the program using => `gcc process.c -o process`
- (5) Execute the program using:- `./process`

Parent and child Process Creation

- > (1) The `fork()` system call is used to create a new process.
- (2) After `fork()`:
 - Parent process receive child PID
 - Child process receives 0

(3) Use `getpid()` to display Process ID.

(4) Use `getppid()` to display Parent Process ID.

Orphan Process

(1) Create a child process using `fork()`.

(2) Terminate the process before the child finishes execution.

(3) The child process becomes an orphan process.

(4) The orphan process is adopted by `init/systemd` process.

Zombie Process

(1) Create a child process using `fork()`.

(2) Terminate the child process.

(3) Parent process does not call `wait()`.

(4) The child remains in zombie state until the parent terminates.