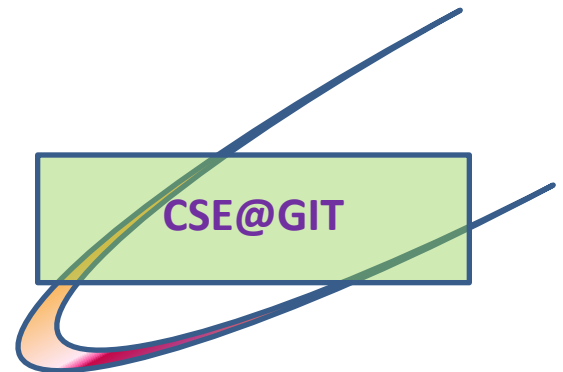


Experiment No. 7

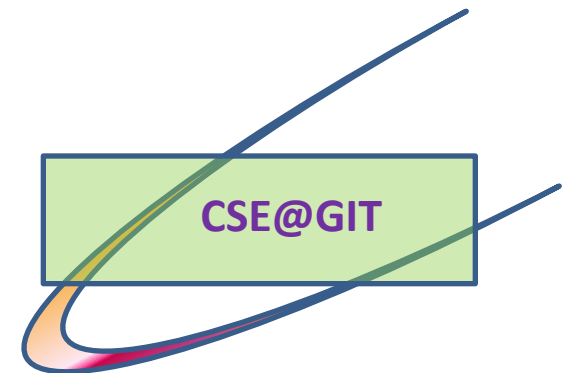
Problem Definition:

Write a C/C++ program that creates a zombie and then calls system to execute the ps command to Verify that the process is zombie.



Objectives of the Experiment:

- 1) To understand the concept of Zombie process.
- 2) To familiarize the creation of Zombie process.
- 3) To understand the concept of process status.



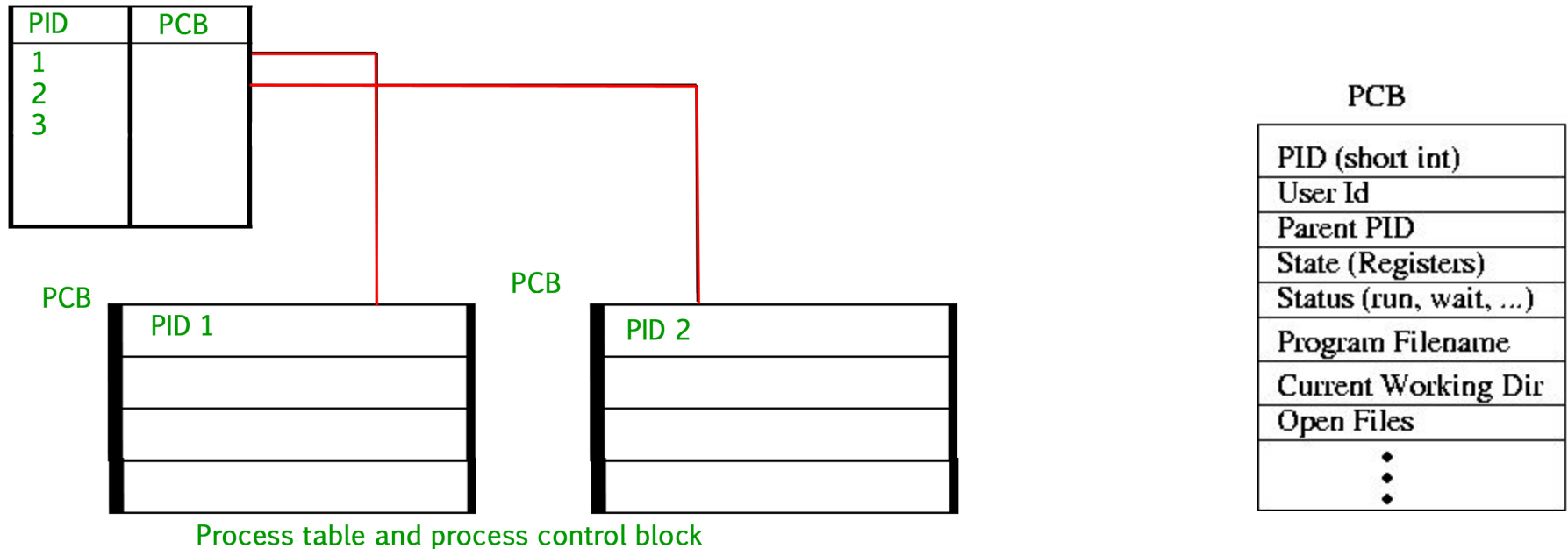
Background of the Experiment

- Process Table
- Process State

• Process Table

- The process table is **simply a data structure in the RAM of a computer.**
- It holds information about the processes that are currently handled by the OS.
- This information includes general information about each process. Like: process id, process owner, parent of process, etc.

Process Table



- **Process Status**

- In most Unix and Unix-like operating systems, the **ps** command **displays the currently-running processes**.
- A related Unix utility named top provides a real-time view of the running processes.

```
[root@localhost ~]# ps
  PID TTY          TIME CMD
 22227 pts/0        00:00:00 bash
 22327 pts/0        00:00:00 man
 22349 pts/0        00:00:00 sh
 22351 pts/0        00:00:00 less
 22566 pts/0        00:00:00 ps
[root@localhost ~]# █
```

```
[root@localhost ~]# ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
root         22227 22224  0 00:28 pts/0        00:00:00 bash
root         22327 22227  0 00:35 pts/0        00:00:00 man ps
root         22349 22327  0 00:35 pts/0        00:00:00 sh -c /usr/bin/bzip2 -c -d /var/cache/man/cat1/ps.1.bz2 | /usr/bin/less -is
root         22351 22349  0 00:35 pts/0        00:00:00 /usr/bin/less -is
root         22570 22227  0 01:10 pts/0        00:00:00 ps -f
[root@localhost ~]# █
```

```
[root@localhost ~]# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0  2136   664 ?        Ss   Feb08   0:02 init [5]
root           2  0.0  0.0     0     0 ?        S    Feb08   0:01 [migration/0]
root           3  0.0  0.0     0     0 ?        SN   Feb08   0:00 [ksoftirqd/0]
root           4  0.0  0.0     0     0 ?        S    Feb08   0:00 [watchdog/0]
root           5  0.0  0.0     0     0 ?        S    Feb08   0:03 [migration/1]
root           6  0.0  0.0     0     0 ?        SN   Feb08   0:00 [ksoftirqd/1]
root           7  0.0  0.0     0     0 ?        S    Feb08   0:00 [watchdog/1]
root           8  0.0  0.0     0     0 ?        S<   Feb08   0:01 [events/0]
root           9  0.0  0.0     0     0 ?        S<   Feb08   0:00 [events/1]
root          10  0.0  0.0     0     0 ?        S<   Feb08   0:00 [khelper]
```

- **Process States**

- Each process in UNIX can be in either of the states:

- **S** - sleep

- **T** - terminated

- **R** - running

- **Z** - zombie

What is Zombie Process???

- On Unix and Unix-like computer operating systems, a zombie process is **a process that has completed execution (via the exit system call) but still has an entry** in the process table.

How does a process becomes Zombie ???

- A process which has finished the execution but still has entry in the process table to report to its parent process is known as a zombie process.
- A **child process** always **first becomes** a **zombie** before **being removed** from the **process table**.

How to create a Zombie process??

- if you want to create a zombie process,
 - First create a **child process**,
 - after the `fork()` , the child-process **should exit()** , and
 - the parent-process should **sleep()** before exiting,

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main ()
{
    pid_t child_pid; /* Create a child process. */
    child_pid = fork ();

    if (child_pid == 0)
    {
        exit (0); /* This is the child process. Exit immediately. */
    }
    else
    {
        sleep(3); /* This is the parent process. Sleep for a minute. */
        system("ps -e -o pid,ppid,stat,cmd");
    }
    return 0;
}
```

OUTPUT

```
22224    1 Sl  gnome-terminal
22226 22224 S   gnome-pty-helper
22227 22224 Ss  bash
22327 22227 T   man ps
22349 22327 T   sh -c /usr/bin/bzip2 -c -d /var/cache/man/cat1/ps.1.bz2 | /usr/bin/less -is
22351 22349 T   /usr/bin/less -is
22585 22227 S+  ./a.out
22586 22585 Z+  [a.out] <defunct>
22587 22585 R+  ps -e -o pid,ppid,stat,cmd
[root@localhost ~]#
```



ZOMBIE PROCESS

Learning Outcomes of the Experiment

At the end of the session, students should be able to :

- 1) Create a Zombie process[L3].
- 2) Explain the process status of all processes.[L2].

Thank You