

## **Programming Problems : 3.20**

**Purpose :** Implement pid manager for managing process identifiers.

**Input :** none

**Output :** none

**OS :** Linux

### **Capabilities :**

Implemented the following API for obtaining and releasing a pid:

- `int allocate_map(void)`—Creates and initializes a data structure for representing pids; returns—1 if unsuccessful, 1 if successful
- `int allocate_pid(void)`—Allocates and returns a pid; returns— 1 if unable to allocate a pid (all pids are in use)
- `void release_pid(int pid)`—Releases a pid

define a structure containing pid and bitmap to store pid situation that a bitmap in which a value of 0 at position i indicates that a process id of value i is available and a value of 1 indicates that the process id is currently in use.

```
os@debian: ~/Desktop
File Edit View Terminal Help
os@debian:~/Desktop$ ./output
pid : 2383
pid : 0
3.19.c          OUTPUTshell
3.20.c          outshell
book1.c         sample.c
cdrom           Screenshot-createProcess.c (~~Desktop) - gedit.png
cdrom0          Screenshot-os@debian: ~~Desktop-1.png
create_2pro-1.c Screenshot-os@debian: ~~Desktop-2.png
create_2pro.c   Screenshot-os@debian: ~~Desktop.png
createProcess.c Screenshot-os@debian: ~.png
createProcessChild shell1.png
createProcessChild.c shell2.png
createProcessFork.c shell.c
exitProcess.c   temp.c
out             unix_shell_1.png
OUTcreate_2pro  Unix_Shell.c
output          zombie.png
Child Complete
os@debian:~/Desktop$ ./output
pid : 2385
pid : 0
3.19.c          OUTPUTshell
3.20.c          outshell
```

By Mina Shaigan