

# Unix Assignment7

Team19 108072244 邱煒甯 108032053 陳凱揚 112065525 簡佩如

## Implementation

Initially, We use `fork()` to create a child process. In this child process, `setsid()` is utilized to create a new session. Subsequently, We verify that the child is a process group leader and no longer has a controlling terminal by using `ps` command to print PID, PGRP, and TPGID. Concurrently, in the parent process, we call `wait(NULL)` to wait the completion of the child process.

```
7  pid_t pid = fork();
8
9  // Check if fork() returns an error
10 if (pid < 0) {
11     perror("Fork failed");
12     exit(1);
13 }
14 // Parent process
15 else if (pid > 0) {
16     printf("Parent process pid: %d\n", getpid());
17     // Parent waits for the child
18     wait(NULL);
19 }
20 // Child process
21 else {
22     printf("Child process pid: %d\n", getpid());
23     // Create a new session
24     if (setsid() == -1) {
25         perror("setsid failed");
26         exit(1);
27     }
28     // Verify that the child is a process group leader and has no controlling terminal
29     char command[100] = {};
30     sprintf(command, "ps -o pid,pgid,tpgid -p %d", getpid());
31     system(command);
32 }
```

## Question

- Explain why the child process does not have a controlling terminal.

When `setsid()` is invoked, the child process becomes the session leader of a new session, the process group leader of a new process group, and is disassociated from its controlling terminal. This is a part of the POSIX standard for session management, ensuring that a session leader that created a new session does not inherit a controlling terminal from the parent process. This

avoids receiving signals from the keyboard (like interrupt and quit) that are intended for the terminal's foreground process group.

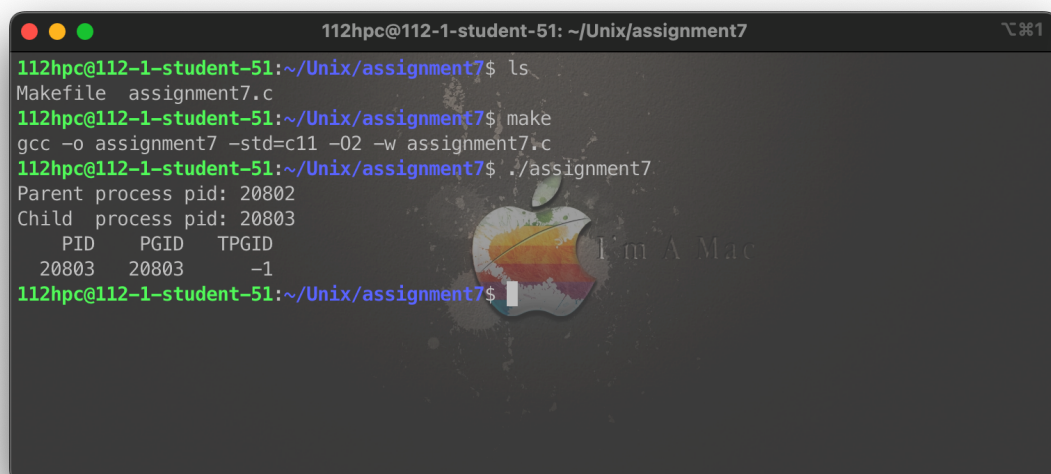
- What are PID, PGRP, and TPGID values in this case? What are their meanings?

PID (Process ID) is the unique identifier for a process.

PGRP (Process Group ID) is the identifier for a group of processes. A process group is a collection of one or more processes, usually associated with the same job, that can receive signals collectively.

TPGID (Terminal Process Group ID) is the process group ID of the process group that owns the terminal. If the value is -1, the process group has no controlling terminal, which will be the case for the child process after `setsid()`.

## Result

A terminal window titled '112hpc@112-1-student-51: ~/Unix/assignment7' with standard macOS window controls. The terminal shows the following commands and output:

```
112hpc@112-1-student-51:~/Unix/assignment7$ ls
Makefile assignment7.c
112hpc@112-1-student-51:~/Unix/assignment7$ make
gcc -o assignment7 -std=c11 -O2 -w assignment7.c
112hpc@112-1-student-51:~/Unix/assignment7$ ./assignment7
Parent process pid: 20802
Child process pid: 20803
  PID  PGID  TPGID
  20803 20803   -1
112hpc@112-1-student-51:~/Unix/assignment7$
```

The output indicates that a child process was successfully created with its own PID (20803) and PGRP (20803), and it no longer has a controlling terminal (TPGID -1).

PID	PGID	TPGID
20803	20803	-1