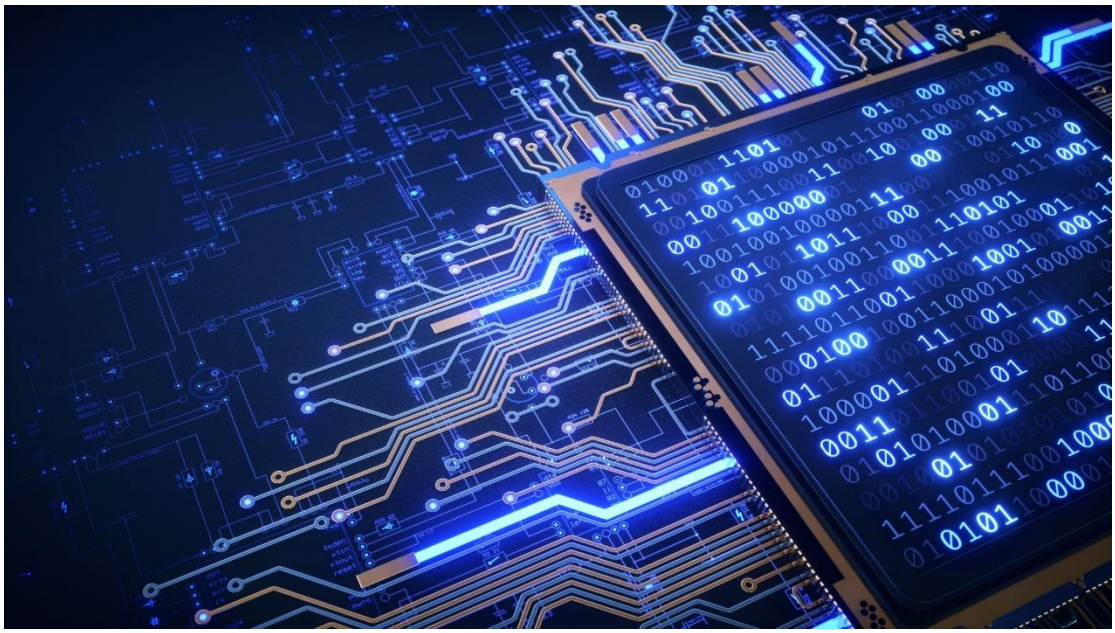
ADVANCED UNIX PROGRAMMING ASSIGNMENT REPORT

ASSIGNMENT 7



TEAM 9 — 林禾堃、馬毓昇、陳曦

Nov 2023

1. Code Implementation

First, define the command for later use:

```c
#include<unistd.h>
#include<stdio.h>
#include<errno.h>
#include<stdlib.h>
#include<sys/wait.h>
#define PSCMD "ps -o pid,pgid,tpgid -p "
extern int errno;
```

Then, we fork a new child by **fork()**:

```c
int main(){
        pid_t pid;
        char cmd[100];
        if((pid = fork()) < 0){
                // some error occured
                perror("Error occured while running fork: ");
                return errno;
        }
```

After that, we create a new session in the child process: by **setsid()**:

```c
        else if(pid > 0){
                // parent
                //printf("\nOridinal Status: \n");
                //sprintf(cmd, "%s %d\0", PSCMD, getpid());
                //system(cmd);
        }
        else{
                // child
                sleep(1);
                pid = setsid();
                //printf("\nNew Status: \n");
                sprintf(cmd, "%s %d\0", PSCMD, getpid());
                system(cmd);
                exit(0);
        }
```

In it, we print out the PID, PGID and TPGID by predefined command.

2. Result Screen Shot

```
root@genet0:~/Advanced-UNIX-Programming/HW7 # ./assignment7
  PID  PGID TPGID
17839 17839     0
```

As this shows, since the PID and PGID are the same, we can tell that this process is the process group leader. Additionally, since the TPGID is 0, it doesn't have a controlling terminal.

3. Answer to the Question

3.1 Why the child process doesn't have a controlling terminal

According to bear's slide,

:

If the calling process is not a process group leader, this function creates a new session

- The process becomes the session leader of this new session
- The process is the only process in this new session
- The process becomes the process group leader of a new process group.
- The new process group ID is the process ID of the calling process
- The process has no *controlling terminal*

Since we made the child the process group leader by **setsid()**, it doesn't have a controlling terminal.

3.2 The PID, PGID, TPGID values and their meanings

	value	meaning
PID	17839	These two values are the same, indicating that this process is the process group leader.
PGID	17839	
TPGID	0	The process doesn't have a controlling terminal