

LAB 4 : Processes

Objective

- To learn process creation and management in Linux.

Instructions

1. Please do the following exercises and for each question produce a file for the code and screen shot of the output.
2. Show your work to your TA.
3. Collect the screen shots in a PDF file and upload All code files and the PDF file on [TEAMS](#) before the end of the lab to get full marks.
4. Use the following file naming convention for the PDF file:

[LAB4_section_your-first-name_student-id](#)

Note: 1 mark (out of 10) deducted per day late. 3 days late maximum.

Exercises

1. (a) Modify `code1.c` program such that it will create 8 processes **and** print the process id along with the message. What is the process id of the parent (the first process) and the last created child?

(b) How many processes will be created by the following program?

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #define MAX_ITER 2
4
5 int main () {
6
7     for (int i = 1; i <= MAX_ITER; i++) {
8         fork();
9     }
10    return 0;
11 }
```

(c) Include the statement `printf ("Welcome to Madinah! My pid is %d. \n", getpid());` right after `fork()` and run the program. Does it give you the same output as question 1(a)? Explain the output. Note: refer to the Tree of Processes discussed in the slide.

2. Given the following program:

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main () {
5     char *programName = "ps";
6     char *pathName = "/bin/ps";
7
8     printf ("Before calling exec() \n");
9     execlp (pathName, programName, NULL);
10    printf ("After calling exec() \n");
11
12    return 0;
13 }
```

- a. Compile and run the program. Explain your output. Is the message After calling exec() being displayed? Why?
- b. Modify the above program so that it executes the executable of question 1a above and display the process ID. Why are the PIDs the same?

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main () {
5     char *programName = "Q1a";
6     char *pathName = "/home/user/Q1a";
7
8     printf ("Before calling exec(). My pid is %d \n", getpid());
9     execlp (pathName, programName, NULL);
10    printf ("After calling exec(). My pid is %d \n", getpid());
11
12    return 0;
13 }
```

Note: Change the variable `pathname` with the proper path.

3. Write and run program `fork.c` (discussed in the lecture).

- a. Explain the output.
- b. Remove the statement `wait (NULL);` and run the program. Is there any different in the output? Why?

4. Compile and run the following program. Why are the values different?

```
1 ▾ #include <stdio.h>
2   #include <unistd.h>
3   #include <sys/types.h>
4
5   int value = 100;
6
7 ▾ int main () {
8
9     pid_t pid = fork();
10
11 ▾    if (pid==0) {
12        value += 10;
13    }
14
15 ▾    else if (pid>0) {
16        value -= 5;
17    }
18    printf ("value = %d. My pid = %d\n", value, getpid());
19
20    return 0;
21 }
22
```