

**SCR2043 OPERATING SYSTEMS**

Name : MUHAMMAD AMIRUL FAHMI BIN NOOR ANIM

Marks

Student ID : B19EC0018

Section : 02

Example: An execution of program 1 produces a process tree, an output, and process identifier assignment as shown in Figure 1. The parent process identifier is assumed to have been assigned with an integer 21, while the child is assigned with the next available identifier i.e 22.

```
# include <stdio.h>
# include <unistd.h>
main() {
    int pid1;
    printf(" \n Hello World\n");
    pid1=fork();
    if (pid1 < 0 )
        printf("\n Error in forking \n")
    else if (pid1 == 0)
        printf("\n Knock! Knock! Knock!\n");
    else (pid1 > 0)printf("\n Please come in \n");
}
```

program 1

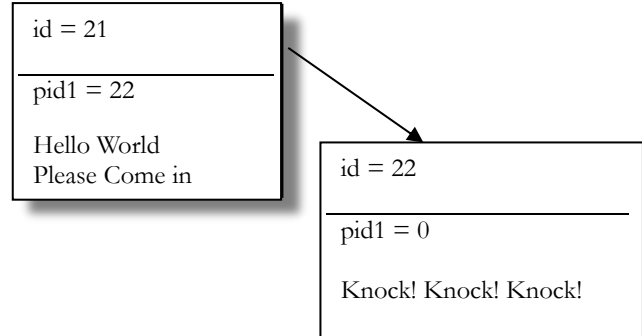


Figure 1: Process tree of program 1

Please take note: Students are not allowed to run the code for Q1 on computer.

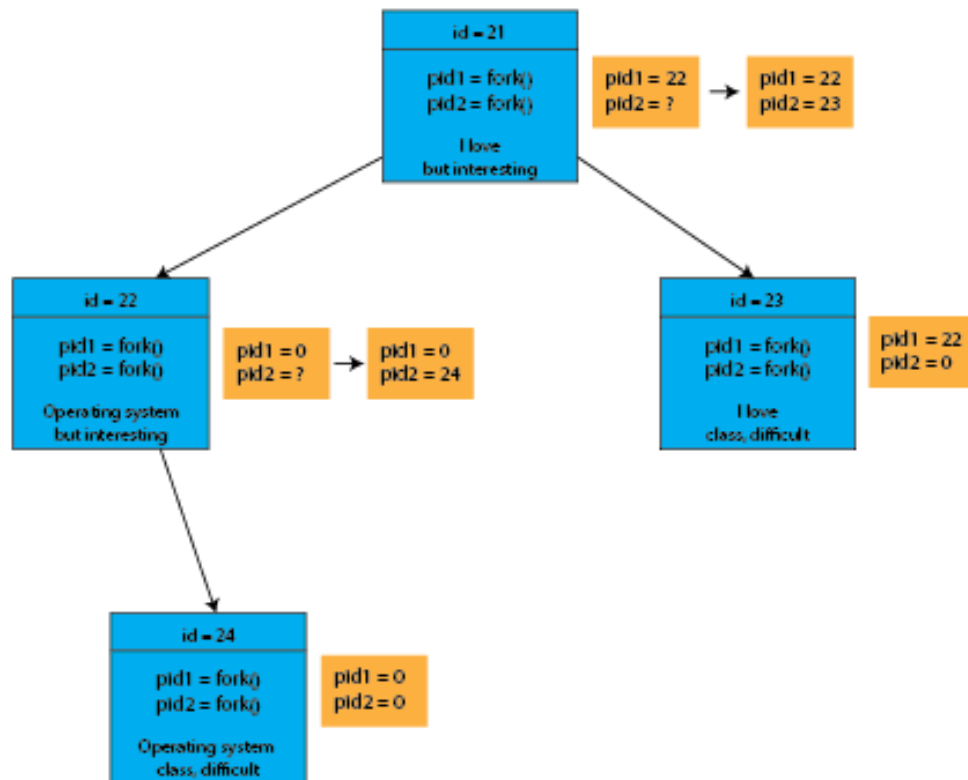
- Q1 : Draw a process tree (as in Figure 1) to illustrate parent-child relationship for the C code given in program 2. For each node on the tree, write the following information:
- The process identifier (assumed that the parent process identifier is assigned with an integer 21).
  - The return value of the fork() statement
  - Generated output

```
# include <stdio.h>
# include <unistd.h>

main() {
    int pid1, pid2;
    pid1=fork();
    pid2=fork();
    if (pid1 == 0) {
        printf("\n Operating system \n");
    } else {
        printf("\n I love \n");
        if (pid2 == 0) {
            printf("\n class, difficult \n");
        } else
            printf("\n but interesting!!\n");
    }
}
```

Program 2

Answer Q1:



Q2: Referring to Figure 2, write a program that will create four processes – original, clone1, clone2, clone3, and clone4. Each process displays message that illustrate the parent - child/children relationship between processes.

- i. Process *original* will display an output as follow:
 

“ I am the original process, my identifier, my clones are:  
 process clone1: *id\_of\_clone1*  
 process clone2: *id\_of\_clone2*”  
 process clone3: *id\_of\_clone3*”
- ii. Process *clone1* will display an output as follow:
 

“I am the obsolete version, my identifier is *id\_of\_clone1*”
- iii. Process *clone4* will display the following message:
 

“ I am most sophisticated, my parent is clone1 with an identifier of *id\_of\_clone1* ”

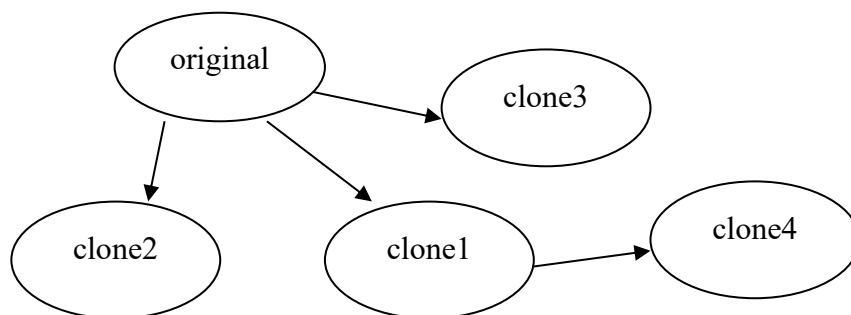


Figure 2: Tree of five processes

Write down your code on the space provided. Compile your code on Linux environment and determine the identifier of the following processes by fill in the following blanks:

- a. original process    0
- b. clone1 process    1
- c. clone2 process    2
- d. clone3 process    3
- e. clone4 process    4

Code for Figure 2:

```
#include <stdio.h>
#include <unistd.h>

main(){
int i, pid;
for (i=0; i<3; i++) {
pid[i] = fork();
if (pid[i]< 0) {
printf("Sorry, cannot fork\n");
}
else if (pid[i] ==0) {
printf ("I am the obsolete version, my identifier is \n", pid[i]);
}
else if (pid[i] ==4) {
printf ("I am most sophisticated, my parent is clone1 with an identifier of \n", pid[i]);
}
else {
printf("I am the original process, my identifier, my clones are: \n ");
printf("process clone1: ", pid(1));
printf("process clone2: ", pid(2));
printf("process clone3: ", pid(3));
}}
}
```