SCR2043 OPERATING SYSTEMS
Lab 2

Name : _____

Student ID : _____ Section : _____

REFER CHAPTER 3 PROCESS ANIMATION

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

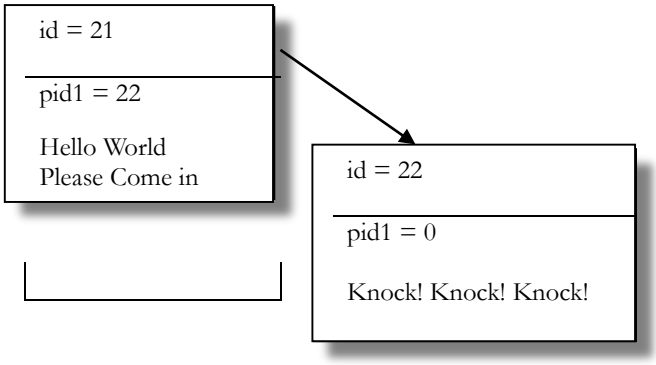

Figure 1: Process tree of program 1

Q1 : Firstly, understand the coding and rewrite back using UNIX program.
Then, draw a process tree (as in Figure 1) to illustrate the parent-child relationship for the C code given in program 2.
For each node on the tree, write the following information:

i. The process identifier (assumed that the parent process identifier is assigned, **id=21**).
ii. The return value of the `fork()` statement
iii. Generated output

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

main()
{
        int pid1, pid2, pid3;
        pid1=fork();
        if (pid1 == 0)
        printf("\n I love \n");
        else
        {
        printf("\n Operating system\n");
        pid2=fork();
        if (pid2 == 0)
        printf("\n class, \n");
        else
        {
        printf("\n difficult \n");
        pid3=fork();
        printf("\n but interesting!!\n");
        }
        }
}
```

Program 2: Generation of child process

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

i. Process *original* will print output as follows:
"I am the PARENT, my process identifier is *id_parent*, my clones are: "
process clone1: *id_child1*
process clone3: *id_child3*"

ii. Process *clone1* will print output as follow:
"I am the obsolete VERSION; the recent version is *id_child2"*

iii. Process *clone4* will print the following message:
"I am the most SOPHISTICATED CLONE, process identifier of my parent is *id_child3 "*

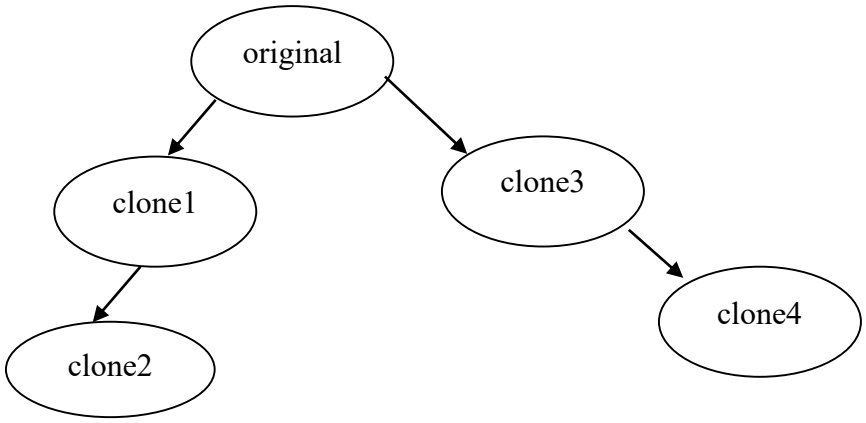

Figure 2: Tree of five processes