King Saud University

College of Computer and Information Sciences

CSC 227: Operating Systems

Homework 2: Processes

Due Date: Thursday 8 November 2018

Part I:

Consider this C program:

```
#include <stdio.h>
#include <stdlib.h>
int y = 7;
int main(int argc, char *argv[])
{
     int *values;
     int i;
     values = (int *) malloc (sizeof (int)* 5);
     for (i=0; i < 5; i++)
          values[i] = i;
     return 0;
}
```

When we run this program, the process will be loaded in Main Memory, in which section of the process will the following be stored:

1- The variable y:

2- The parameters argc, *argv:

3- The pointer *values:

4- The actual data that the pointer *values points to (values):

5- The variable i:

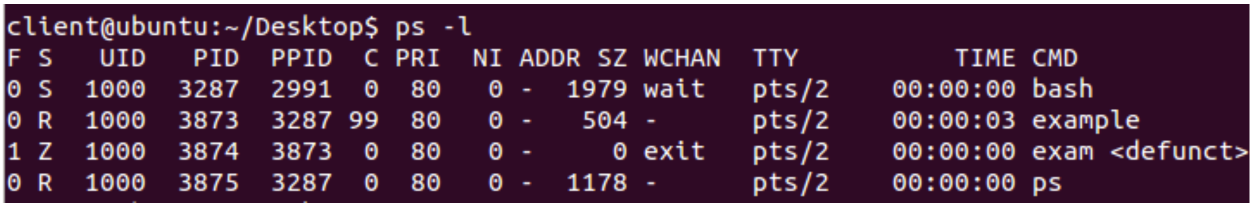
Part II:

Using the Linux VM you created in Homework 1, write a C program that forks a child process that ultimately becomes a zombie process.

Use the command ***ps*** to obtain the states of the processes in your system.

> The easiest way to check the state of your child process is to run your program in the background using the operand ***&,*** for example, let’s say you named your program “example”, you could run it in the background by executing:
>
> ***./example &***
>
> You can then execute the command “***ps -l***” to show the list of processes run through your terminal.

You should obtain a screenshot like the one below. Notice, the state of the child process (PID: 3874) is ***Z*** under the column S. Z means zombie.



```
client@ubuntu:~/Desktop$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  3287  2991  0  80   0 -  1979 wait   pts/2    00:00:00 bash
0 R  1000  3873  3287 99  80   0 -   504 -      pts/2    00:00:03 example
1 Z  1000  3874  3873  0  80   0 -     0 exit   pts/2    00:00:00 exam <defunct>
0 R  1000  3875  3287  0  80   0 -  1178 -      pts/2    00:00:00 ps
```

1) Provide in your answer the source code, a screenshot showing the commands you used to obtain the state of the child process, and a screenshot of the state of the parent and child processes.
2) Examine the man page of the command ***ps***. Find the option that shows you the tree process of the entire OS.
    - Draw the portion of the process tree that shows the tree path from the child process upto the **init** process. In each node of the tree provide the **name** and ***PID*** of the process.
    - Provide a screenshot showing your processes (parent & child) in the process tree.
    - Use the command: ***kill -9 PIDofParentProcess*** to kill the parent process. Did it kill the zombie child, too? Show a screenshot showing whether or not the zombie child was removed.