Wayne State University

CSC 4421 - Winter 2020
Computer Operating Systems Labs
Lab 03 - Process Control I

Instructor 001: Rui Chen

Points Possible: 100

Tasks

Task 1* (20 points)

Here is a C program that prints out the process id, the parent process id, the process user id, and group id. Please type it in your computer and save as task1.c. Now compile it and run it to see how it works.

Include your screenshots in your submission.

```
#include <stdio.h>
#include <unistd.h>
int main (void)
{
  printf("I am process %ld\n", (long) getpid());
  printf("My parent is %ld\n", (long)getppid());
  printf("Process user id is %ld\n", (long)getuid());
  printf("Group id is %ld\n", (long)getgid()); return 0;
}
```

Task1.c

Task 2* (80 points)

To create a new process, you can use the system call 'fork(void)'. Incomplete code is given below. The fork(void)system call is used to create a child process. Replace the ______ with proper code. It should work as follows.

- For the parent process, the program will print out "Parent Process: Global variable: 4 Function variable: 22";
- For the child process, it will print out "Child Process: Global variable: 3 Function variable: 21".

```
#include <sys/types.h>
#include"stdio.h"
#include <unistd.h>
int globalVariable = 2;
int main()
{
  char parentStr[] = "Parent Process";
  char childStr[] = "Child Process";
  char *string = NULL;
  int functionVariable = 20;

  // Create a child process.
  pid_t pid = ______;

  if (______) // Failed to fork
  {
    perror("Unable to create child process");
    return 1;
  }

  else if (______) // child
  {
    // Code only executed by child process
    string = &childStr[0];
    globalVariable++;
    functionVariable++;
  }

  ______    // parent
  {
    // Code only executed by parent process
    string = &parentStr[0];
    globalVariable += 2;
    functionVariable += 2;
  }

  // Code executed by both parent and child.
  printf("%s\n", string);
  printf(" Global Variable: %d\n",globalVariable);
  printf(" Function Variable: %d\n", functionVariable);
}
```

Task2.c