National University

of Computer & Emerging Sciences-Lahore

Operating Systems
Assignment # 2
Assignment deadline Friday, 24th February 2023, 11:00 PM
Read Guidelines carefully

Question # 01

Write a program that creates a child process using fork ().The child displays its ID and parent ID. The Parent displays its own ID and its child ID. The output should be of the form:

I am parent my ID is=1234 and my Child ID is=4567
I am Child my ID is=4567 and my Parent ID is=1234

Question # 02

Develop a stopwatch with lap counter and total time calculator. The program will create two processes and they will start calculating time. One process will be used to calculate total time and the other will calculate lap time. The total time calculator process will keep on calculating time. In the lap time, calculator process when it reaches lap time limit, the process will display lap number and set its counter to zero and start calculating again. Take input of lap time and number of laps from the user and start the program. At the end, display total time from total time calculator process.

Question # 03

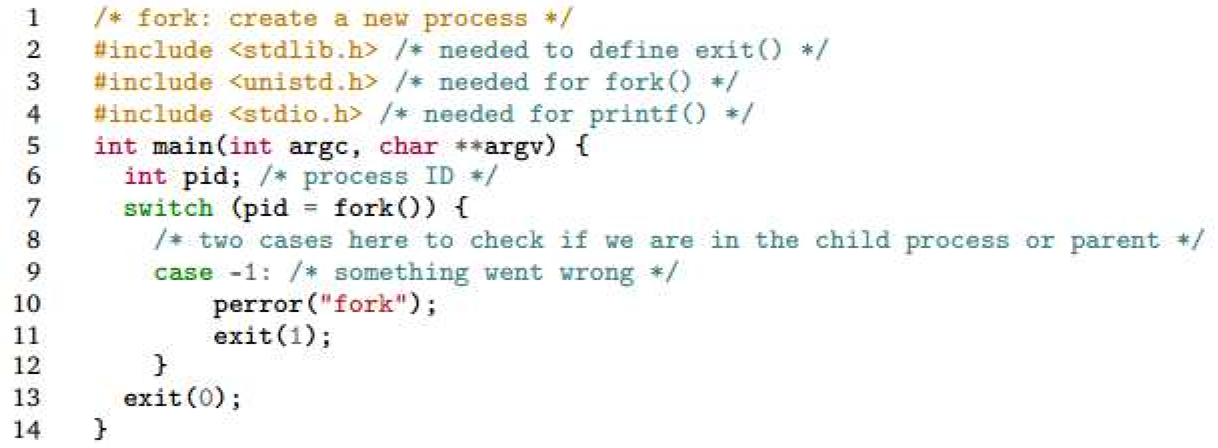
Why we use vfork ()? Write difference between fork() and vfork()?

Question # 04

Write a C++ program in which a parent process creates a child process using fork () system call. The child takes a number as input and print its factorial while the parent waits for the child to terminate using wait (). After the child exit () the parent displays the exit status returned by the child and call exit ().

Question # 05

(a) Write another program to issue the fork system call. Complete the following code for the provided requirements.



```
/* fork: create a new process */
#include <stdlib.h> /* needed to define exit() */
#include <unistd.h> /* needed for fork() */
#include <stdio.h> /* needed for printf() */
int main(int argc, char **argv) {
  int pid; /* process ID */
  switch (pid = fork()) {
    /* two cases here to check if we are in the child process or parent */
    case -1: /* something went wrong */
        perror("fork");
        exit(1);
    }
  exit(0);
}
```

(b) Complete the two cases above. One should execute only in child process and should print "In child" and the child's process ID. The second should run only if we are in parent and print both the parent's and the child's PID.

Question # 06

Debug and compile the following piece of codes by including appropriate header files and write the output.

a)

```
char *mesag= "This is a message ";
int main();
{
char buf[1024];
int fd[2];
pipe(fd);
if(fork()!=0){/* I am parent */
writes(fd[1], mesag, strlen(mesag) + 1);
}
else {/* child */
read(fd[0],buf,1024);
printf(Got this from MaMa!!!\n%s\n, buf);
}
return 0;
}
```

b)

```
 int main()
 {
 int pfds[2];
```

```
pipes(pfds);
if (!fork())
{
close(1);
dupl(pfds[1]);
close(pfds[0]);
execlpz("who",  who, NULL);
}
else
{
close(0);
dup(pfds[0]);
closes(pfds[1]);
execlp("wc", "wc", "-l", NULL);
}
}
```

Question # 7

Write a program for inter-process communication among three processes using pipes. In which, one process is Parent which has two child processes. Parent process gets a number from the user. First child reads the number from the pipe and calculates the factorial. Similarly, the second child reads the number from pipe and calculates the square of that number. In the end, the Parent process read the output from the pipe and display it.

Guidelines:

- A single violation of guideline will lead to **Zero mark** in your assignment.
- Deadlines should be kept in mind. **No extension** in assignment dates would be given.
- Submit the ***.c*** or ***.cpp*** file and the output's screenshot in **Word** file on **Google classroom in Zip form**.
- You are encouraged to take help from Internet and books.
- You will have maximum marks if you have done the entire task.
- This is an individual assignment. **PLAGARISM IS NOT ACCEPTABLE!**
- Follow the instructions as it is, otherwise, your assignment would not be accepted at all.