LAU
School of Arts and Sciences

Lebanese American University
School of Arts and Sciences
Department of Computer Science and Mathematics

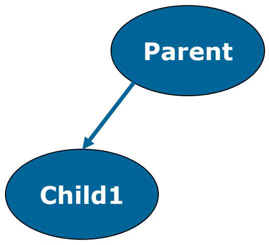
CSC326 – Operating Systems
Dr. Abdallah DAbboussi

Lab Assignment 5

Due: Sunday Oct 23, 2022

Problem 1:

a. Write a C program that will create the following processes and in every process display the PID and the parent's PPID.

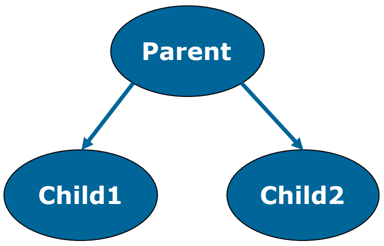


Sample Output:

```
Child1 Process PID= 1190, PPID=1189
Parent Process PID= 1189, PPID=1140
```

- *Submit your solution in a file called "Problem1a.c".*

b. Write a C program that will create the following processes and in every process display the PID and the parent's PPID.

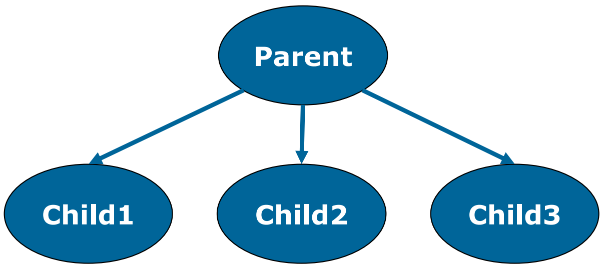


Sample Output:

```
Child1 Process PID= 1205, PPID=1204
Child2 Process PID= 1206, PPID=1204
Parent Process PID= 1204, PPID=1140
```

- *Submit your solution in a file called "Problem1b.c".*

c. Write a C program that will create the following processes and in every process display the PID and the parent's PPID.

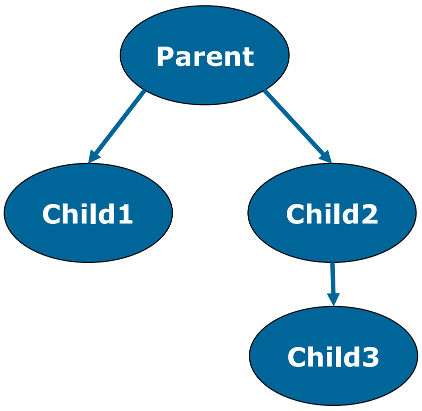


Sample Output:

```
Child1 Process PID= 1224, PPID=1223
Child2 Process PID= 1225, PPID=1223
Child3 Process PID= 1226, PPID=1223
Parent Process PID= 1223, PPID=1140
```

- *Submit your solution in a file called "Problem1c.c".*

d. Write a C program that will create the following processes and in every process display the PID and the parent's PPID.



Sample Output:

```
Child1 Process PID= 1268, PPID=1267
Child3 Process PID= 1270, PPID=1269
Child2 Process PID= 1269, PPID=1267
Parent Process PID= 1267, PPID=1140
```

- *Submit your solution in a file called "Problem1d.c".*

Problem 2:

Write a C program that:

1. Displays the PID and the PPID of the main parent process
2. Forks a child
3. Checks if the fork system call failed, raise an error and exit the program
4. The child should first display its PID and its parent PPID, and then display the current date using the execlp function call
5. The parent should wait for the child to terminate. The parent should first display its PID, child PID and its parent PPID. The parent then should retrieve the exit status of its child and displays it. In case the execlp failed, the parent should display a corresponding message such as "Child Failed" with the child status, otherwise "Child Succeeded".

Sample Output:

```
Main Process PID=1208, PPID=1114
Child Process PID= 1209, PPID=1208
Sat 02 Oct 2021 05:11:23 PM EDT
Parent Process PID= 1208, child PID= 1209, PPID=1114
Child exit code=0
Child succeeded!
```

- *Submit your solution in a file called "Problem2.c".*

Problem 3:

Write a C program that:

1. Displays a message containing the PID and the PPID of the main process
2. Creates a child process using the fork() system call
3. The child should first sleep for 5 sec
4. The child should then display "Ready to kill my parent!"
5. The child should kill its parent via a signal (use *kill (parent_id, SIGINT)*).
6. The child should display "I am an orphan now"
7. The parent should wait for the child and display "I am the parent"

Sample Output:

```
Main Process PID= 1664, PPID=1114
Child Process PID= 1665, PPID=1664
Ready to kill my parent!
I am an orphan now
```

- *Submit your solution in a file called "Problem3.c".*

Problem X:(Optional)

Write a C program that:

1. Prints a message containing the PID and the PPID of the process
2. Forks a child. You should check if the fork call failed, raise an error and exit the program
3. Prints a message containing the PID and the PPID of the child's process
4. Calls fork from the child and exits if fork fails
5. Prints a message containing the PID and the PPID of the grandchild's process
6. Exits the grandchild process. (Print a message stating that the process is exiting)
7. Prints a message containing its PID and its PPID in the child process. The child process should wait for the grandchild to terminate and print a message that it is exiting and exists
8. Prints a message containing its PID and its PPID in the parent process. Let the process sleep for 2 seconds and then use execvp or execlp to execute the "ps" command

Sample Output:

```
Original process with PID 1584 and PPID 1114
Child process with PID 1585 and PPID 1584
Parent process with PID 1584 and PPID 1114
Child process with PID 1585 and PPID 1584
Grand child process with PID 1586 and PPID 1585
Grandchild is about to exit
Child is about to exit
Parent is about to execute the ps command
PID TTY          TIME CMD
1114 pts/0    00:00:00 zsh
1584 pts/0    00:00:00 ps
```

- *Submit your solution in a file called "ProblemX.c".*