LAB NO 04

THE EXEC FUNCTION

Fall 2024

CSE-302L Systems Programming Lab

Submitted by:

Name: **Hassan Zaib Jadoon**

Reg no**: 22PWCSE2144**

Class Section **: A**

Signature: _____

Submitted to:

Engr. Abdullah Hamid

December 29, 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

LAB NO 04 THE EXEC FUNCTION

Task 1

Write a program that takes N UNIX commands as arguments, creates N child processes, each of them implementing their respective commands. Parent process shall wait for all the child processes and receive and print the exit status of the child processes.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
 int cmnds = argc - 1;
 pid_t pid;
 for (int i = 1; i <= cmnds; i++) {
        pid = fork();
 if (pid < 0) {
    perror("Fork failed");
  return -1;
        } else if (pid == 0) {
 execlp(argv[i],argv[i],NULL);
 perror("execlp failed");
 return -1;
        }
else {
    int status;
wait(&status);
if(WIFEXITED(status)){
printf("Child exited with status %d\n",WEXITSTATUS(status));
}
}
 } return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s: ~/Desktop/SP Lab/SP Lab 04
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 04$ ./task1.o ls pwd ps
task1add.png  task2add.c  task2full.png  task3max.c     task3minmax.o
task1.c       task2add.o  task2mul.c     task3max.o     task3minmaxo.png
task1.o       task2a.png  task2mul.o     task3max.png   task3minmax.png
task1o.png    task2b.png  task2mul.png   task3min.c     task3min.o
task1.png     task2.c     task2.o        task3minmax.c  task3min.png
Child exited with status 0
/home/hassan/Desktop/SP Lab/SP Lab 04
Child exited with status 0
  PID TTY          TIME CMD
 4651 pts/4    00:00:00 bash
 4699 pts/4    00:00:00 task1.o
 4702 pts/4    00:00:00 ps
Child exited with status 0
```

Task 2

a) Write a program that takes integers as arguments and adds them.
b) Write a program that takes integers as arguments and multiplies them.
c) Write a program that takes integers as arguments & adds & multiplies them using the above two programs.

Code:

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
 int sum = 0;
 for (int i = 1; i < argc; i++) {
   sum += atoi(argv[i]);
  }
 return sum;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
 int product = 1;
 for (int i = 1; i < argc; i++) {
  product *= atoi(argv[i]);
    }
 return product;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>

int main (int argc, char*argv[]){
  pid_t pid1,pid2;
pid1=fork();
if(pid1==0){
execv("./task2add.o", argv);
perror("Exec Failed");
}
pid2=fork();
if(pid2==0){
execv("./task2mul.o", argv);
perror("Exec Failed");
}
int status;
waitpid(pid1,&status, 0);
if(WIFEXITED(status)){
printf("Addition: %d\n", WEXITSTATUS(status));
}
waitpid(pid2,&status,0);
if(WIFEXITED(status)){
printf("Multiplication %d\n", WEXITSTATUS(status));
}
return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 04$ ./task2.o 3 5 4 3
Addition: 15
Multiplication 180
```

Task 3

Write a program “minmax.c” that takes an array as command line arguments. Program executes min.c and max.c programs in its two child processes. One child process calculates and returns the min value and other calculates and returns the max value in the array. The program “minmax.c” shall receive the values returned by the child processes and display these values.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
 int max = atoi(argv[1]);
 for (int i = 2; i < argc; i++) {
 int num = atoi(argv[i]);
 if (num > max) {
  max = num;
   }
 } return max;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
 int min = atoi(argv[1]);
  for (int i = 2; i < argc; i++) {
   int num = atoi(argv[i]);
 if (num < min) {
   min = num;
        } }
return min;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
  int x = fork();
  if(x == 0){
    execv("task3min.o", argv);
  }
  int y = fork();
  if(y == 0){
    execv("task3max.o", argv);
  }
   int status;
   waitpid(x, &status, 0);
   if (WIFEXITED(status)) {
        printf("Min value: %d\n", WEXITSTATUS(status));
  }
    waitpid(y, &status, 0);
    if (WIFEXITED(status)) {
        printf("Max value: %d\n", WEXITSTATUS(status));
  }
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 04$ ./task3minmax.o 5 24 3 6
Min value: 3
Max value: 24
```

CSE 302L: SYSTEMS PROGRAMMING LAB

LAB ASSESSMENT RUBRICS

Criteria & Point Assigned	Outstanding 2	Acceptable 1.5	Considerable 1	Below Expectations 0.5	Score
Attendance and Attentiveness in Lab PLO08	Attended in proper Time and attentive in Lab	Attended in proper Time but not attentive in Lab	Attended late but attentive in Lab	Attended late not attentive in Lab	
Capability of writing Program/ Algorithm/Drawing Flow Chart PLO1, PLO2, PLO3, PLO5,	Right attempt/ no errors and well formatted	Right attempt/ no errors but not well formatted	Right attempt/ minor errors and not well formatted	Wrong attempt	
Result or Output/ Completion of target in Lab PLO9,	100% target has been completed and well formatted.	75% target has been completed and well formatted.	50% target has been completed but not well formatted.	None of the outputs are correct	
Overall, Knowledge PLO10,	Demonstrates excellent knowledge of lab	Demonstrates good knowledge of lab	Has partial idea about the Lab and procedure followed	Has poor idea about the Lab and procedure followed	
Attention to Lab Report PLO4,	Submission of Lab Report in Proper Time i.e., in next day of lab., with proper documentation.	Submission of Lab Report in proper time but not with proper documentation.	Late Submission with proper documentation.	Late Submission Very poor documentation	

Instructor:

Name: _____ Signature: _____