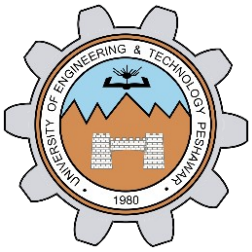
The exec Function

LAB # 04



Fall 2020

CSE302L System Programming Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Madiha Sher

Sunday, December 20th, 2020

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Objective(s):

- Understand and implement exec Function.

Task # 01:

Write a program that takes N UNIX commands as arguments, creates N child processes, each of them implementing their respective commands.

Code:

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

int main(int argc, char *argv[])
{
  int x;
  for(int i=1;i<argc;i++)
  {
    x=fork();
    if(x==0)
    {
      printf("Pid: %d,  Executing: %s\n",getpid(),argv[i]);
      execlp(argv[i],argv[i],NULL);
      break;
    }
  }

}
```

Output / Graphs / Plots / Results:

```
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 1$ ./Task1 ps ls who pwd
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 1$ Pid: 2255,  Executing: pwd
/home/shah/Systems Programming/labs/Lab 4/Task 1
Pid: 2254,  Executing: who
shah     tty7         2020-12-19 18:37 (:0)
Pid: 2253,  Executing: ls
Task1  Task1.c
Pid: 2252,  Executing: ps
   PID TTY          TIME CMD
  2090 pts/0    00:00:00 bash
  2252 pts/0    00:00:00 ps
```

Task # 02:

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.**

Task 2.a:

```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]);
  }
  printf("Total Sum: %d\n",sum);
}
```

Output:

```
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 2$ ./Task2a 7 3 2
Total Sum: 12
```

Task 2.b:

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

int main(int argc, char *argv[])
{
  int prod=1;
  for(int i=1;i<argc;i++)
  {
    prod*=atoi(argv[i]);
  }
  printf("Total Product: %d\n",prod);
}
```

Output:

```
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 2$ ./Task2b 7 3 2
Total Product: 42
```

Task 2.c:

```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>

int main(int argc, char *argv[])
{
  int x;
  for(int i=1;i<3;i++)
  {
    x=fork();
    if(x==0)
    {
      if(i==1)
      {
        strcpy(argv[0],"Task2a");
        execv(argv[0],argv);
      }else{
        strcpy(argv[0],"Task2b");
        execv(argv[0],argv);
      }
      break;
    }
  }

}
```

Output:

```
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 2$ ./Task2 7 3 2
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 2$ Total Product: 42
Total Sum: 12
```

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 processes 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.

Min:

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


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

Max:

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

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

Minmax:

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

int main(int argc,char *argv[])
{
  int value;
  int x;
  for(int i=1;i<3;i++)
  {
    x=fork();
    if(x==0)
    {
      if(i==1)
      {
        strcpy(argv[0],"./min");
        printf("Min:\n");
        execv(argv[0],argv);
        perror("Failed to execute min\n");
      }else {
        strcpy(argv[0],"./max");
        printf("Max:\n");
        execv(argv[0],argv);
        perror("Failed to execute max\n");
      }
      break;
    }
  }
  for(int i=0;i<2;i++)
  {
      int Pid = wait(&value);
      printf("Pid: %d, Status: %d\n",Pid,WEXITSTATUS(value));
  }
}
```

Output:

```
ShahRaza@ubuntu:~/Systems Programming/labs/Lab 4/Task 3$ ./minmax 7 3 8 12 9
Max:
Pid: 2480, Status: 12
Min:
Pid: 2479, Status: 3
```