Name: Prathamesh Sudhir Paraswar

Class:TE9

Batch:L9

Roll No: 33155

Assignment 2B:

Execve: **1) execve**() executes the program pointed to by *filename*.

2) *filename* must be either a binary executable, or a script starting with a line of the form "**#!** *interpreter* [arg]"

3) **execve**() does not return on success, and the text, data, bss, and stack of the calling process are overwritten by that of the program loaded.

4) Arguments: 1) First argument consists of filename to be called.

2) argv: argv is an array of argument strings passed to the new program.

3) envp: envp is an array of strings, conventionally of the form key=value.

Code:

```
# include <stdio.h>

# include <stdlib.h>

# include <unistd.h>

# include<sys/types.h>

# include <sys/wait.h>

void perror(const char * s);

void sort(int arr[],int n)

{

  int i,j,temp;

  for(i=0;i<n;i++)
```

```
  {
    for(j=i+1;j<n;j++)
    {
        if(arr[i] > arr[j])
        {
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
        }
    }
  }
}
int main(int argc, char *argv[])
{
        int n;
        printf("Enter the number of elements\n");
        scanf("%d",&n);
        int arr[n];
        printf("Enter the elements:\n");
        for(int i=0;i<n;i++)
        {
                scanf("%d",&arr[i]);
        }
```

```
printf("The array before sorting is:\n");
for(int i=0;i<n;i++)
{
	printf("%d",arr[i]);
}
printf("\n");
int k=fork();
if(k==0)
{
	printf("\nInside the child process\n");
	sort(arr,n);
	printf("The array after sorting is:\n");
	for(int i=0;i<n;i++)
	{
		printf("%d",arr[i]);
	}
	printf("\n");
	char *newargv[n+2] ;
	for(int i=0;i<=n;i++)
	{
		newargv[i]=(char *)malloc(5*sizeof(char));
		sprintf(newargv[i],"%d",arr[i]);
	}
```

```
        newargv[n+1]=NULL;
        char *newenviron[] = { NULL };
        execve("./myecho.o",newargv,newenviron);
        printf("Hi");
    }
    else if(k>0)
    {
        printf("Inside the parent Process\n");
        wait(NULL);
        printf("\nParent Process is completed\n");
    }
}
```