OPERATING SYSTEMS

LAB-5

COE18B023
G. HRUTHIK

1. CODE

```
 #include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
void reverse(char str[], int n)
{
   char temp;
   for(int i=0; i<=n/2 ;i++)
   {
        temp = str[i];
        str[i] = str[n-i];
        str[n-i]=temp;
   }
}
int main()
{
   int fd1[2],fd2[2];
   char writestr[20],readstr[20];
   printf("Enter the string :- ");
   gets(writestr);
   pid_t pid;
   if(pipe(fd1)==-1) {
```

```
        fprintf(stderr, "Pipe 1 failed");
        return 1;
    }
    if(pipe(fd2)==-1) {
        fprintf(stderr, "Pipe 2 failed");
        return 1;
    }
    pid = fork();
    if(pid==-1)
    {
        printf("Fork failed");
        return 1;
    }
    if(pid>0)
    {
        close(fd1[0]);
        close(fd2[1]);
        write(fd1[1], writestr, strlen(writestr)+1);
        read(fd2[0], readstr, sizeof(readstr));
        printf("\nPARENT BLOCK : String after reversing(from child) -->
%s\n",readstr);
    }
    else {
        close(fd1[1]);
        close(fd2[0]);
        read(fd1[0],readstr, sizeof(readstr));
        printf("\nCHILD BLOCK : String to be reversed(from parent) -->
%s \n",readstr);
        reverse(readstr,strlen(readstr)-1);
        write(fd2[1], readstr, strlen((readstr))+1);
        printf("\nCHILD BLOCK : String after reversing --> %s
\n",readstr);
    }
    return 0;
}
```

OUTPUT :

```
hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$ gcc 1.c
1.c: In function ‘main’:
1.c:21:2: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
   21 |  gets(writestr);
      |  ^~~~
      |  fgets
/usr/bin/ld: /tmp/ccEhv1g8.o: in function `main':
1.c:(.text+0xb5): warning: the `gets' function is dangerous and should not be used.
hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$ ./a.out
Enter the string :- hruthik

CHILD BLOCK : String to be reversed(from parent) --> hruthik

CHILD BLOCK : String after reversing --> kihturh

PARENT BLOCK : String after reversing(from child) --> kihturh
hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$
```

2. CODE

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
int main()
{
   int fd1[2], fd2[2];
   pid_t pid;
   char str1[20], str2[20],readstr[20];
   printf("Enter string 1 : ");
   gets(str1);
   printf("Enter string 2 : ");
   gets(str2);
   if(pipe(fd1)==-1)
   {
         fprintf(stderr, "Pipe 1 failed");
         return 1;
   }
   if(pipe(fd2)==-1)
   {
         fprintf(stderr, "Pipe 2 failed");
         return 1;
```

```
    }
    pid = fork();
    if(pid==-1)
    {
        printf("Fork failed");
        return 1;
    }
    else if(pid>0)
    {
        close(fd1[1]);
        close(fd2[0]);
        read(fd1[0], readstr, sizeof(readstr));
        printf("\nPARENT BLOCK: Concatenating the two strings '%s' and
'%s'(received from child)\n", str1, readstr);
        strcat(str1, readstr);
        write(fd2[1], str1, strlen(str1)+1);
    }
    else {
        close(fd1[0]);
        close(fd2[1]);
        write(fd1[1], str2, strlen(str2)+1);
        read(fd2[0], readstr, sizeof(readstr));
        printf("\nCHILD BLOCK: received concatenated string from parent :
%s\n",readstr);
    }
}
```

OUTPUT :

```
hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$ gcc 2.c
2.c: In function ‘main’:
2.c:12:2: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
   12 |  gets(str1);
      |  ^~~~
      |  fgets
/usr/bin/ld: /tmp/ccEDaLQQ.o: in function `main':
2.c:(.text+0x39): warning: the `gets' function is dangerous and should not be used.
hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$ ./a.out
Enter string 1 : Gurugubelli
Enter string 2 : Hruthik

PARENT BLOCK: Concatenating the two strings 'Gurugubelli' and 'Hruthik'(received from child)

CHILD BLOCK: received concatenated string from parent : GurugubelliHruthik
hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$
```

3. CODE

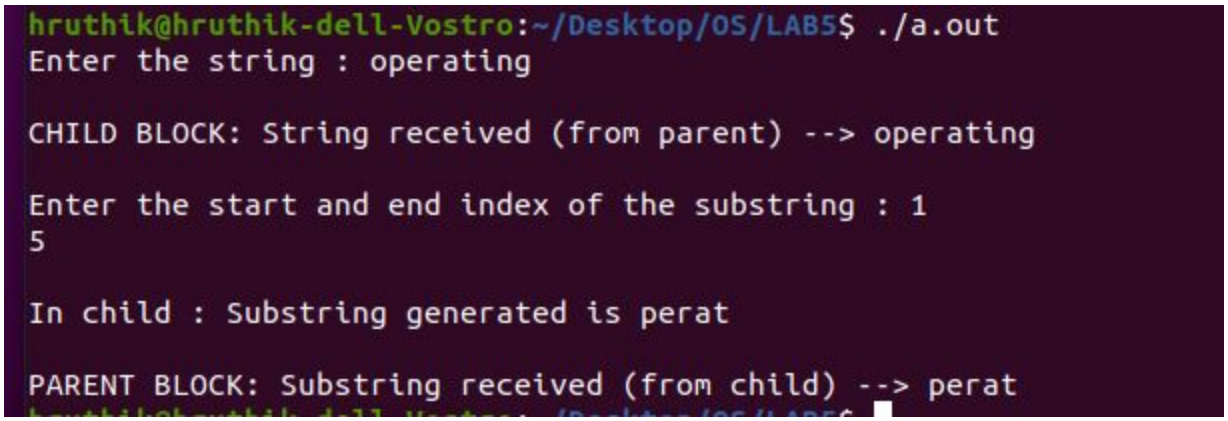
```
 #include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#define MAX 50
int main()
{
   int fd1[2], fd2[2];
   pid_t pid;
   char str[MAX], substr[MAX], readstr[MAX];
   int start,end;
   printf("Enter the string : ");
   gets(str);
   if(pipe(fd1)==-1) {
         fprintf(stderr, "Pipe 1 failed");
         return 1;
   }
   if(pipe(fd2)==-1) {
         fprintf(stderr, "Pipe 2 failed");
         return 1;
   }
   pid = fork();
   if(pid==-1)
   {
         printf("Fork failed");
         return 1;
   }
   if(pid>0)
   {
         close(fd1[0]);
         close(fd2[1]);

         write(fd1[1], str, strlen(str)+1);
         read(fd2[0], readstr, MAX);

         if(strlen(readstr)!=1)
```

```
                printf("\nPARENT BLOCK: Substring received (from child) -->
%s\n",readstr);
    }
    else
    {
        close(fd1[1]);
        close(fd2[0]);
        read(fd1[0], readstr, MAX);
        printf("\nCHILD BLOCK: String received (from parent) --> %s\n",readstr);
        printf("\nEnter the start and end index of the substring : ");
        scanf("%d %d",&start,&end);
        if(start>=0 && start<strlen(readstr) && end>=0 && end<strlen(readstr))
        {
                int j=0;
                for(int i=start; i<=end; i++)
                        substr[j++] = str[i];
                printf("\nIn child : Substring generated is %s\n",substr);
                write(fd2[1], substr, strlen(substr)+1);
        }
        else {
                printf("Index out of range, exiting\n");
                exit(1);
    }
    return 0;
    }
}
```

OUTPUT :



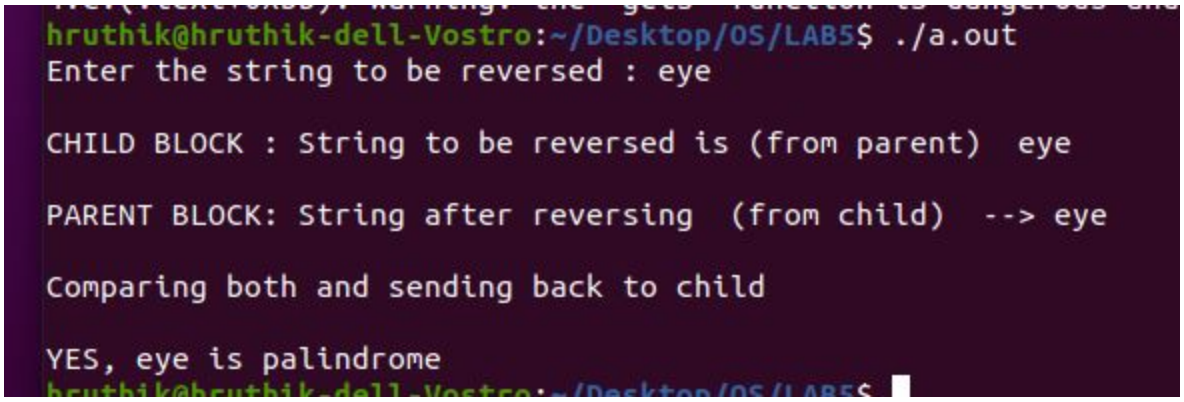
4. CODE

```
 #include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#define MAX 50
void reverse(char str[], int n)
{
   char temp;
   for(int i=0; i<=n/2 ;i++)
   {
         temp = str[i];
         str[i] = str[n-i];
         str[n-i]=temp;
   }
}
int main()
{
   int fd1[2],fd2[2];
   char writestr[MAX],readstr[MAX], result[MAX];
   printf("Enter the string to be reversed : ");
   gets(writestr);
   pid_t pid;
   if(pipe(fd1)==-1) {
         fprintf(stderr, "Pipe 1 failed");
         return 1;
   }
   if(pipe(fd2)==-1) {
         fprintf(stderr, "Pipe 2 failed");
         return 1;
   }
   pid = fork();
   if(pid==-1)
   {
         printf("Fork failed");
```

```
        return 1;
    }
    if(pid>0)
    {
        close(fd1[0]);
        close(fd2[1]);
        write(fd1[1], writestr, strlen(writestr)+1);
        read(fd2[0], readstr, MAX);
        printf("\nPARENT BLOCK: String after reversing  (from child)  -->
%s\n\nComparing both and sending back to child\n",readstr);
        /*if(strcmp(writestr,readstr)==0)
            printf("\nYES!, %s is palindrome\n",writestr);
        else
            printf("\nNO!, %s is not a palindrome\n",writestr);*/
        if(strcmp(writestr,readstr)==0)
            strcpy(result, "YES");
        else
            strcpy(result, "NO");
        write(fd1[1], result, strlen(writestr)+1);
    }
    else {
        close(fd1[1]);
        close(fd2[0]);
        read(fd1[0],readstr, MAX);
        printf("\nCHILD BLOCK : String to be reversed is (from parent)  %s
\n",readstr);
        reverse(readstr,strlen(readstr)-1);
        write(fd2[1], readstr, strlen((readstr))+1);
        read(fd1[0],result, MAX);
        //printf("\n%s",result);
        if(strcmp("YES",result)==0)
            printf("\n%s, %s is palindrome\n",result,writestr);
        else if(strcmp("NO",result)==0)
            printf("\n%s!, %s is not a palindrome\n",result,writestr);

    }
    return 0;
}
```

OUTPUT :


hruthik@hruthik-dell-Vostro:~/Desktop/OS/LAB5$ ./a.out
Enter the string to be reversed : eye
CHILD BLOCK : String to be reversed is (from parent) eye
PARENT BLOCK: String after reversing (from child) --> eye
Comparing both and sending back to child
YES, eye is palindrome