Operating Systems Exercise 5 Pipes

(1) Parent sets up a string which is read by child, reversed there and read back the parent. &
(4) String reversal and palindrome check using pipes / shared memory.
(both combined in 1).

Code:

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

void reverseStr(char *str){
   for (int i = 0; i < strlen(str) / 2; i++){
       char temp = str[i];
       str[i] = str[strlen(str) - i - 1];
       str[strlen(str) - i - 1] = temp;
   }
}


int main(){
   int pipe1[2],pipe2[2];
   int pid;
   int length;
   char inputstr[512];
   printf("Enter the string :\n");
   scanf("%[^\n]%*c",inputstr);
   char readmessage[512];
   char writemessage[512];
   int returnstatus1, returnstatus2;
   returnstatus1 = pipe(pipe1);
   if (returnstatus1 == -1){
       printf("Unable to create pipe 1 \n");
       return 1;
   }
```

```
    returnstatus2 = pipe(pipe2);
    if (returnstatus2 == -1){
        printf("Unable to create pipe2 \n");
        return 1;
    }

    pid = fork();
    if(pid > 0){
        close(pipe1[0]); //close read of pipe1
        close(pipe2[1]); //close write of pipe2
        write(pipe1[1],inputstr,sizeof(inputstr)+1);
        read(pipe2[0],readmessage,sizeof(readmessage));
        printf("Reversed String is  :%s \n",readmessage);
        length=strlen(readmessage);
        if(strncmp(inputstr,readmessage,length)==0)
        printf("%s is a Palindrome\n",inputstr);
        else printf("%s is NOT a Palindrome\n",inputstr);
    }
    else{
        close(pipe1[1]); // close write of pipe1
        close(pipe2[0]); //close read of pipe2
        read(pipe1[0],writemessage,sizeof(writemessage));
        reverseStr(writemessage);
        printf("%s\n",writemessage);
        write(pipe2[1], writemessage, sizeof(writemessage) + 1);
    }

    return 0;
}
```

Output:

```
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ gcc strrev.c -o strRev
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ ./strRev
Enter the string :
Hello this is Vijay
yajiV si siht olleH
Reversed String is  :yajiV si siht olleH
Hello this is Vijay is NOT a Palindrome
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ ./strRev
Enter the string :
nayan
```

```
nayan
Reversed String is  :nayan
nayan is a Palindrome
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$
```

(2) Parent sets up string 1 and child sets up string 2. String 2 concatenated to string 1 at parent end and then read back at the child end.

Code:

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

int main(){
   int pipe1[2], pipe2[2];
   int pid;
   char string1[512], string2[512] ,destination[512];
   printf("Enter string1 :\n");
   scanf("%[^\n]%*c", string1);
   printf("Enter string2 :\n");
   scanf("%[^\n]%*c", string2);
   strcpy(destination,string1);
   char readmessage[512];
   char readmessage2[512];
   int returnstatus1, returnstatus2;
   returnstatus1 = pipe(pipe1);
   if (returnstatus1 == -1){
       printf("Unable to create pipe 1 \n");
       return 1;
   }
   returnstatus2 = pipe(pipe2);
   if (returnstatus2 == -1){
       printf("Unable to create pipe2 \n");
       return 1;
   }
   pid = fork();
   if (pid>0){
```

```
        close(pipe1[1]); //close write of pipe1
        close(pipe2[0]); //close read of pipe2

        read(pipe1[0], readmessage, sizeof(readmessage));
        strcat(destination,readmessage);
        write(pipe2[1],destination,sizeof(destination)+1);
    }
    else{
        close(pipe1[0]); //close read of pipe1
        close(pipe2[1]); //close write of pipe2
        write(pipe1[1],string2,sizeof(string2)+1);
        read(pipe2[0],readmessage2,sizeof(readmessage2));
        printf("Concatnated string is : %s\n",readmessage2);
    }
}
```

Output:

```
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ ./strConcat
Enter string1 :
Assignment
Enter string2 :
pipes
Concatnated string is : Assignmentpipes
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ ./strConcat
Enter string1 :
Vijay
Enter string2 :
 Meena
Concatnated string is : Vijay Meena
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$
```

(3) Substring generation at child end of a string setup at parent process end.

Code:

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

```c
int main(){
   srand(time(0));
   int pipe1[2], pipe2[2],pipe3[2];
   int returnstatus1, returnstatus2, returnstatus3;
   int pid;
   char inputstring[512],readmessage[512];
   int startindex,endindex,length,start,end;
   printf("Enter the string : ");
   scanf("%[^\n]%*c", inputstring);
   length = strlen(inputstring);
   returnstatus1 = pipe(pipe1);
   if (returnstatus1 == -1){
       printf("Unable to create pipe 1 \n");
       return 1;
   }
   returnstatus2 = pipe(pipe2);
   if (returnstatus2 == -1){
       printf("Unable to create pipe2 \n");
       return 1;
   }
   returnstatus3 = pipe(pipe3);
   if (returnstatus3 == -1){
       printf("Unable to create pipe 3 \n");
       return 1;
   }
   pid = fork();

   if(pid>0){
       close(pipe1[0]); // close read of pipe 1
       close(pipe2[0]); // close read of pipe 2
       close(pipe3[0]); // close read of pipe 3
       startindex = rand() % length/2;
       endindex = length/2 + rand() % length/2;
       printf("\n%d %d\n",startindex,endindex);

       write(pipe1[1],inputstring,sizeof(inputstring));
       write(pipe2[1],&startindex,sizeof(startindex));
       write(pipe3[1],&endindex,sizeof(endindex));
```

```
    }
    else{
        close(pipe1[1]); // close write of pipe 1
        close(pipe2[1]); // close write of pipe 2
        close(pipe3[1]); // close write of pipe 3

        read(pipe1[0],readmessage,sizeof(readmessage));
        read(pipe2[0],&start,sizeof(start));
        read(pipe3[0],&end,sizeof(end));
        printf("Substring : ");
        for(int i =start;i<=end;i++)
        printf("%c",inputstring[i]);
        printf("\n");
    }
    return 0;
}
```

Output:

vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ ./subString
Enter the string : hello this is vijay

8 12
Substring : is is
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$ ./subString
Enter the string : this is a test assignment

10 21
Substring : test assignm
vijay@vijay-desktop:~/Desktop/Operating_Systems-master/Pipes$