OPERATING SYSTEMS

NAME: NAGAVENI L G

SEC:4F

SRN: PES2UG21CS315

Programming Exercise 2

Using pipes, reverse a string.

- One process takes in the string as input and writes it to a pipe
- The other process reads from the pipe and reverses the string.

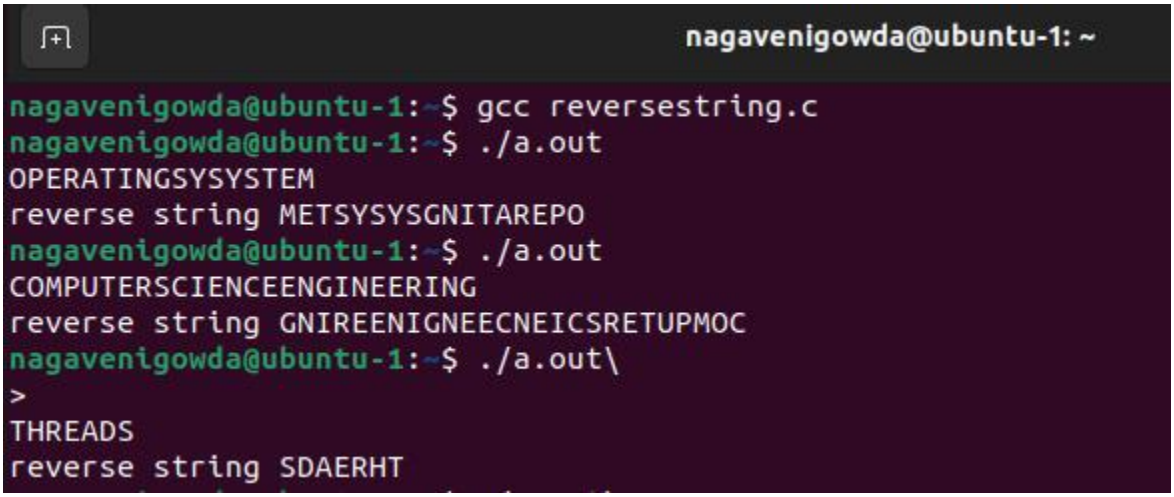
CODE:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
void revstr(char *str1)
{
    // declare variable
    int i, len, temp;
    len = strlen(str1); // use strlen() to get the length of str string

    // use for loop to iterate the string
    for (i = 0; i < len/2; i++)
    {
        // temp variable use to temporary hold the string
        temp = str1[i];
```

```
        str1[i] = str1[len - i - 1];
        str1[len - i - 1] = temp;
    }
}

int main()
{

    // First pipe to send input string from parent


    int fd1[2]; // Used to store two ends of first pipe

    char input_str[100];
    pid_t p;

    if (pipe(fd1) == -1) {
        fprintf(stderr, "Pipe Failed");
        return 1;
    }

    scanf("%s", input_str);
    p = fork();

    if (p < 0) {
        fprintf(stderr, "fork Failed");
        return 1;
    }

    // Parent process
    else if (p > 0) {
        char new_str[100];

        close(fd1[0]); // Close reading end of first pipe

        // Write input string and close writing end of first
        // pipe.
        write(fd1[1], input_str, strlen(input_str) + 1);
        close(fd1[1]);

        // Wait for child to send a string
        wait(NULL);
```

```
        // Read string from child, print it and close
        // reading end.

    }

    // child process
    else {
        close(fd1[1]); // Close writing end of first pipe

        // Read a string using first pipe
        char new_str[100];
        char input_str[100];
        read(fd1[0], input_str, 100);

       revstr(input_str);
        printf("New string %s\n", input_str);
        // Close both reading ends
        close(fd1[0]);




        exit(0);
    }
}
```

OUTPUT:



THANK YOU ☺