Comp 4981 - Assignment 1

Nicole Jingco

A01001875

Overview

This document outlines the requirements, state diagram and pseudocode for the Weird Terminal. The Weird Terminal uses fork, pipes and signals to read, write and process the users input from the terminal. One process will handle the keystrokes, the second process will handle the output prints, and the third process will handle the translation of specified key strokes.

Requirements

The requirements for this program follows the design and functionality listed below.

Constraint

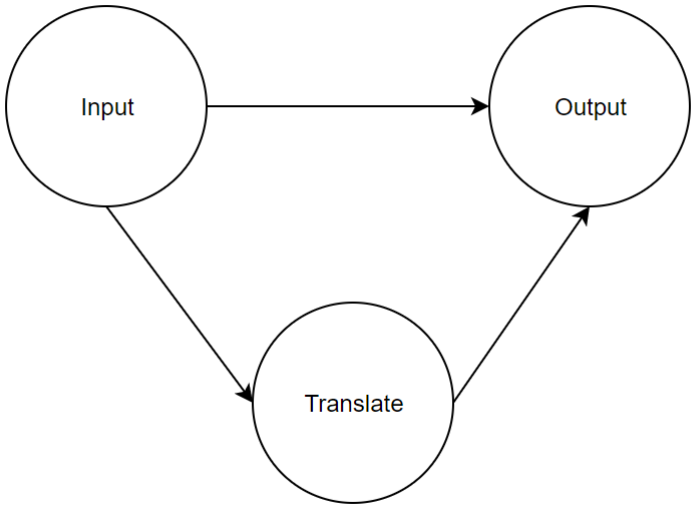
The program requirements must have the following:

- Must be written in C
- Must run on Linux
- Must have three cooperating processes
- Instead of using Linux default values use the following as the commands:
  - "X" - Erase
  - "K" - Line-Kill
  - Control-K - Exit Program (Abnormal termination of program)
  - "T" - Normal Termination
  - "A" - "Z"
  - "a" - "z"
  - "E" - Enter

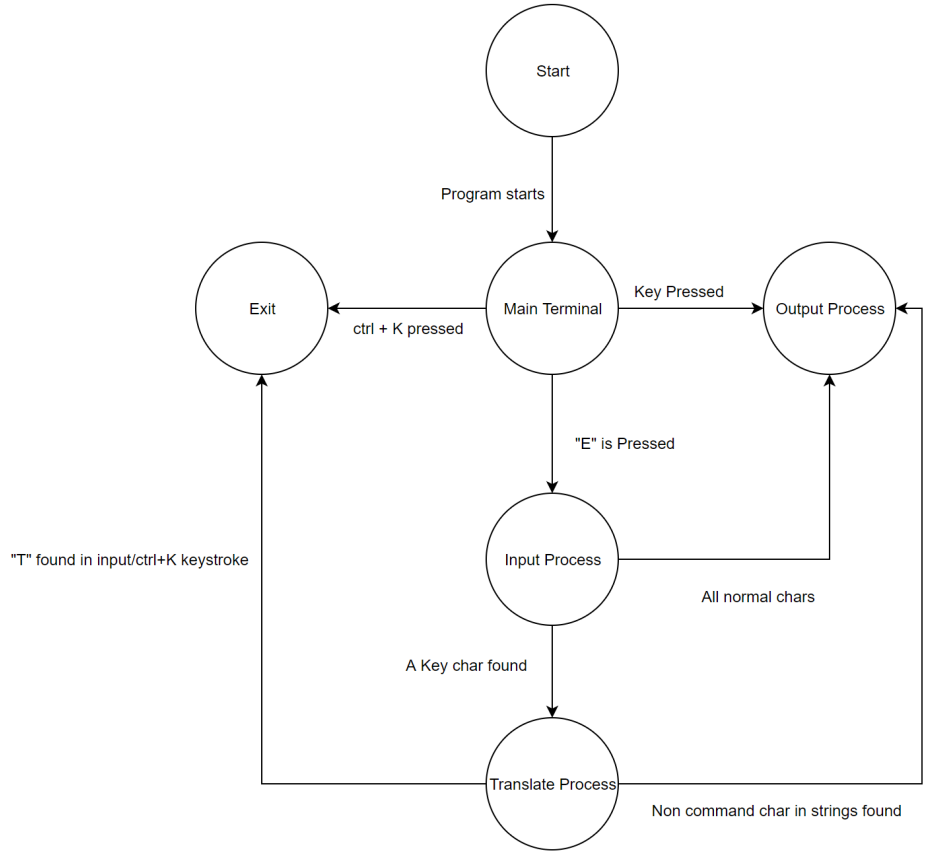
Design

The design requirement must have the following:

- Three different functions are running in separate processes
  1. Input - Reads the characters entered in the keyboard
  2. Translate - converts key characters to its associated character or function
  3. Output - Prints the characters to the terminal
-



State Diagram



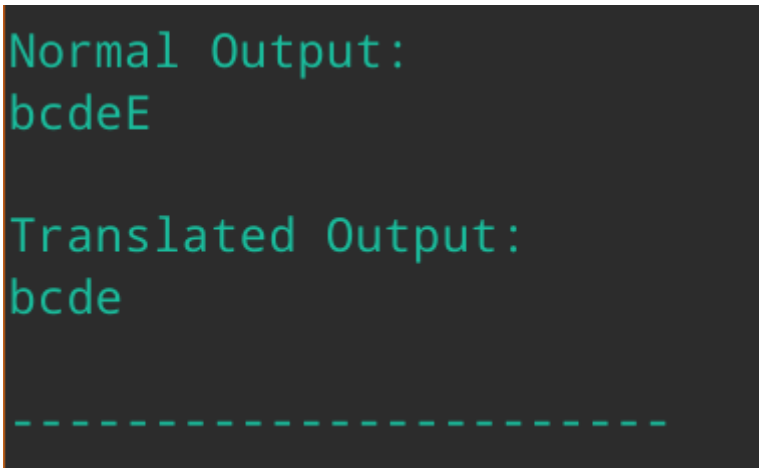
Pseudocode

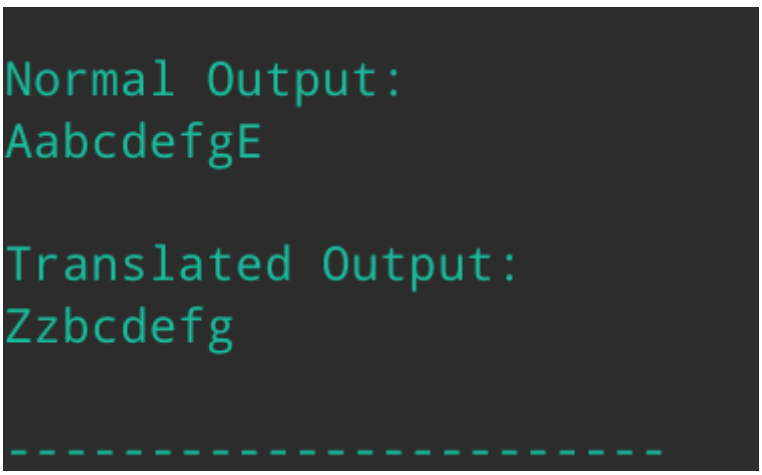
```
// Exit - close the terminal
exit(){
    stop all processes
    close terminal
}

// Main Terminal - read each keystroke and process it if "E" is pressed
```

```
main_terminal(){
    disable keyboard terminal functions

    create pipes for input and translate
    create pipe for input and output
    create pipe for translate and output

    create process for translate
    create process for output

    assign pipes for output process
    assign pipes for translate process
    assign pipes for input process

    kill output output process
    kill translate process

    close pipes
    restore keyboard terminal functions
}

//Input_process - output the strings
input_process(){
    while (there is keystokes)
        store keystroke char to a char buffer array
        if keystorke is ctrl + "k"
            exit
        if keystroke is "E"
            write to translate process
        if keystroke is "T"
            write to output process
            write to translate process
            leave loop
        else
            write to output process
}

// Translate Process - Translate the characters to the assigned
command/translation
tanslate_process(){
    loop through char buffer array
        if char is "A"
            add "Z" to translated array
        else if char is "a"
            add "z" to translated array
        else if char is "X"
            if theres is already a char in translated array
                remove the last char from array
            else
                do nothing
        else if char is "K"
            disregard the characters before 'K' and print the characters after
            break out of loop
        else if char is "T"
            print translated array
            empty char buffer array
            exit
        else
```

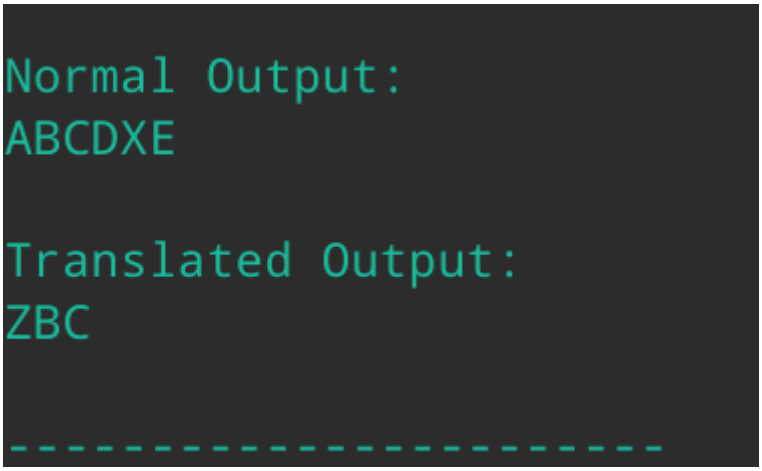
```
            add char to translated array
    print translated array
    empty char buffer array
}

// Output Process - Print the characters to the terminal
output_process(){
    read the char buffer array and print to the terminal
    empty char buffer array
}
```

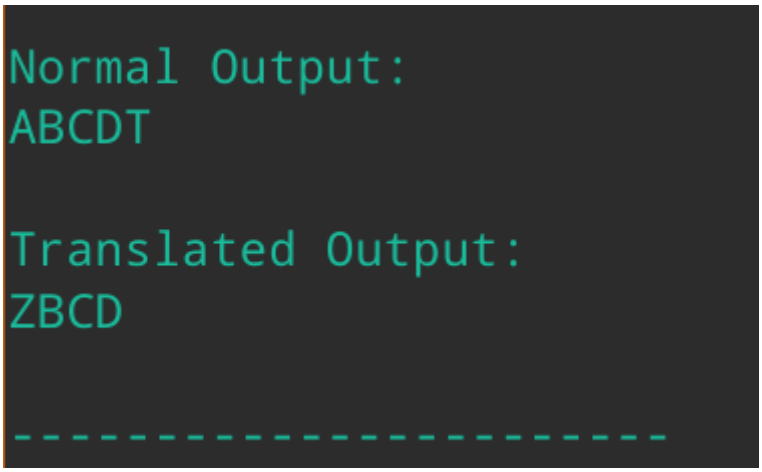
Testing

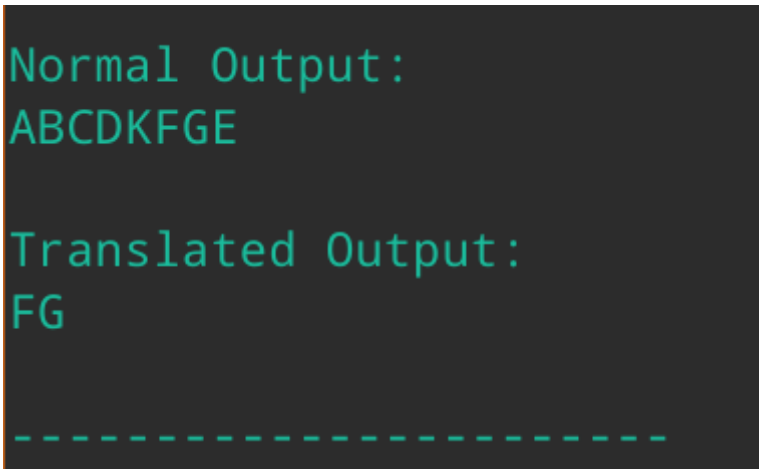
1. No translations
    - Input: "bcdeE"
    - Expected Translation: "bcde"
    - Actual Output:
        - 

2. "A" Translation
    - Input: "AabcefgE"
    - Expected Translation "Zzbcdefg"
    - Actual Output:
        - 

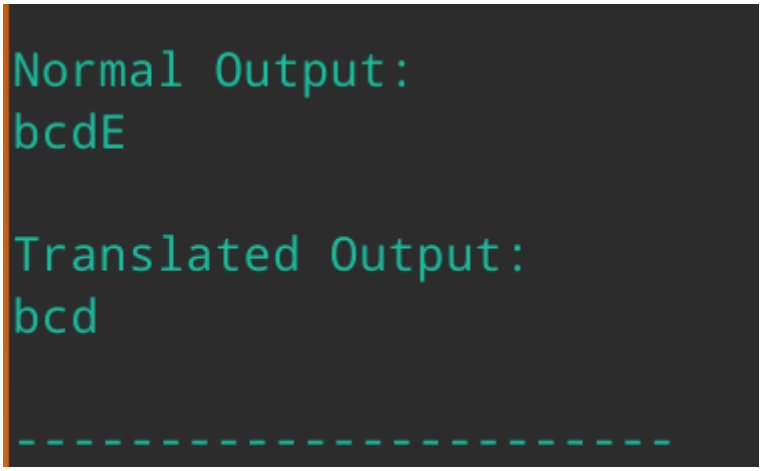
3. "E" Translation
    1. Input: "ABCDXE"

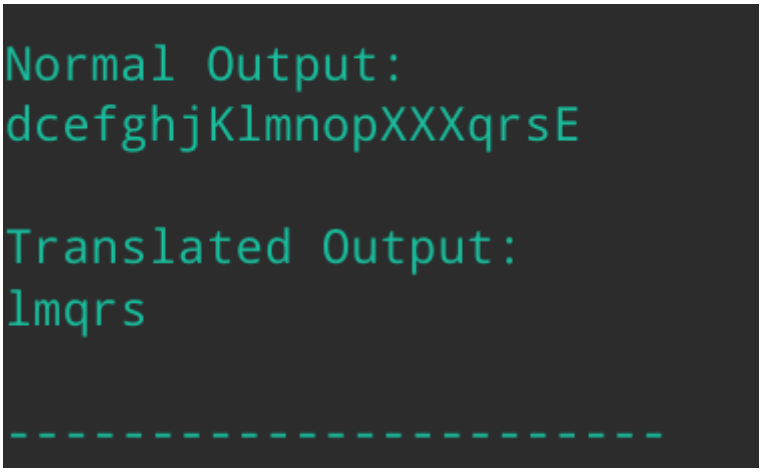
2. Expected Translation: "ZBCD"
3. Actual Output:
    - 

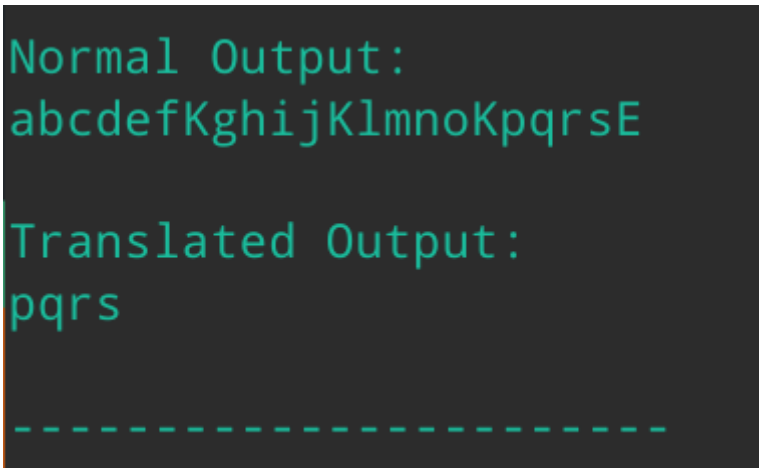
4. "T" Termination
    - Input: "ABCDTE"
    - Expected Translation: "ZBCD" **EXIT**
    - Actual Output:
        - 

5. "K" Translation
    - Input: "ABCDKFGE"
    - Expected Translation: "FG"
    - Actual Output:
        - 

6. No key chars
    - Input: "bcdE"
    - Expected Translation: "bcdE"

- Actual Output:
  - 

7. Combination of 'K' and 'X'
   - Input: "dcefghjKlmnopXXXqrsE"
   - Expected Output: "lmqrsE"
   - Actual Output: "lmqrs"
     - 

8. Multiple 'K' keystrokes
   - Input: "abcdefKghijKlmnoKpqrsE"
   - Expected Output: "pqrs"
   - Actual Output:
     - 

9. Multiple 'X' keystrokes
   - Input: "abcdefghiXXXjklmnoXXpqrXXXs"
   - Expected Output: "zbcdefjklms"
   - Actual Output:

```
Normal Output:
abcdddXXXeeXXXE

Translated Output:
zbc

-----------------------
```

10. Only 'X' input
    - Input: "XXXE"
    - Expected Output:
    - Actual Output:

```
Normal Output:
XXXE

Translated Output:


-----------------------
```

11. Ctrl+K
    - Input: "^K"
    - Expected Output: Exit Program
    - Actual Output:

```

-----------------------

Normal Output:

Killed
[keng@keng-linx LINUX_TERMINAL]$
```