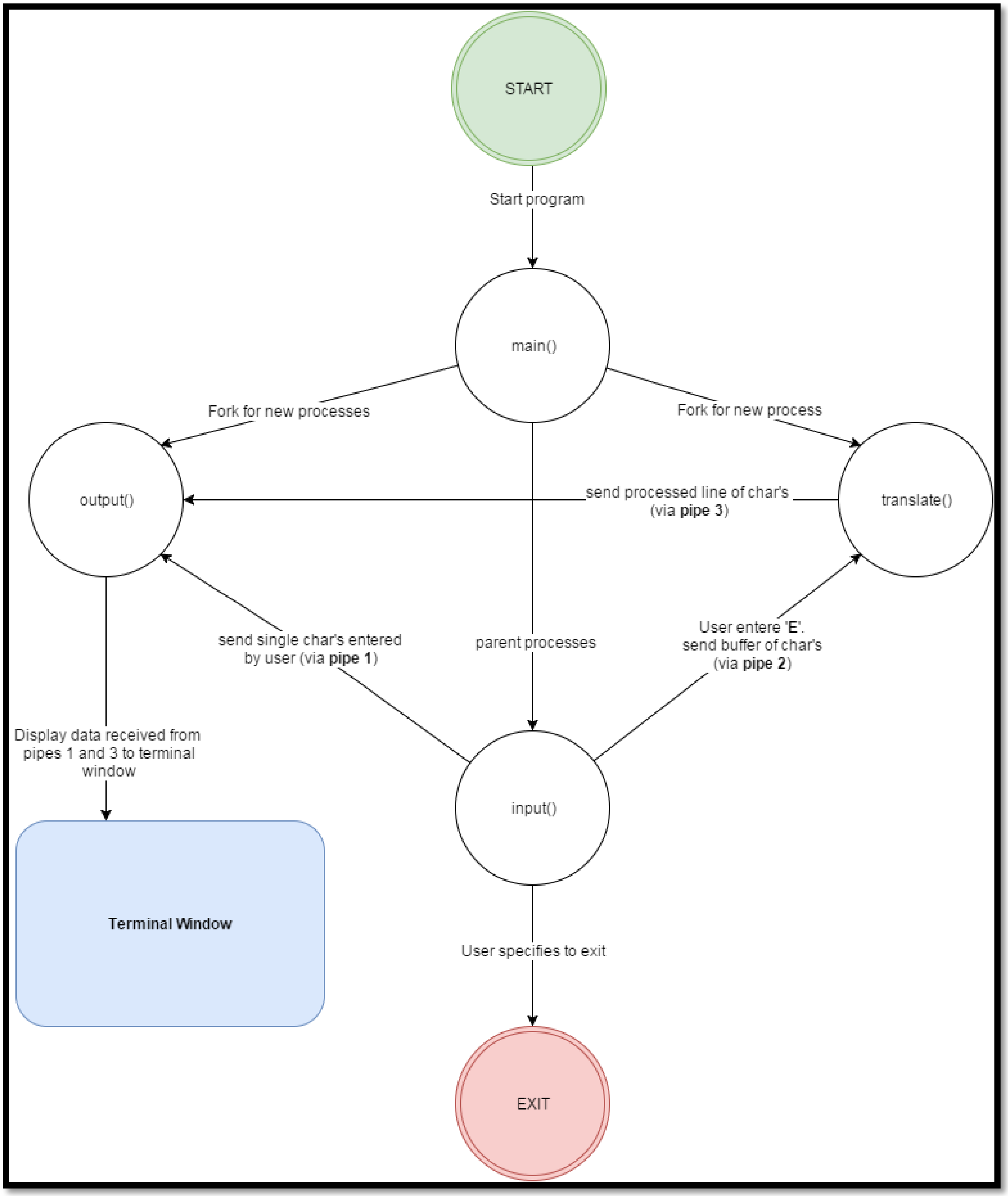
COMP 3981 Assignment 1 Design

Alex Zielinski (A00803488)

1/25/2017

Finite State Diagram


START
Start program
main()
Fork for new processes
Fork for new process
output()
send processed line of char's (via pipe 3)
translate()
send single char's entered by user (via pipe 1)
parent processes
User entere 'E'. send buffer of char's (via pipe 2)
Display data received from pipes 1 and 3 to terminal window
input()
Terminal Window
User specifies to exit
EXIT

User Guide

Terminal settings are disabled so when the user types a keyboard character it will be immediately echoed by the terminal. The user can type the following key (command) characters into the terminal program.

- **E**: acts as the enter button. It processes the buffer created from user input. Any ‘a’ characters will be replaced with ‘z’ characters and will be the character line will be echoed back.
- **K** : acts as a line kill. The moment the user enters the character ‘K’ the buffered user data is sent to translate, the buffer is cleared and an empty line is echoed back to the user.
- **X :** acts as a backspace key. The user can enter an X to indicate a backspace. Once the user enters a **‘E’** key (acts as enter) the data is sent to translate where the any character preceding an ‘X’ is deleted. Also any ‘a’ characters will be replaced with ‘z’ characters
- **T**: acts as a normal terminate key. The moment the user enters this key the program will terminate
- **Ctrl + K** : acts as an interrupt and will “abnormally” terminate the program

Pseudo Code

Program Entry

```
Disable Linux terminal settings
Open 3 pipes
Fork from main processes
If process is child then call output function
If process is parent then fork again
If process is child then call translate function
If process is parent then call input function
Kill processes
Return 0
```

Input Function

```
Close reading from pipe 1 and 2
Go into while loop (while user has not specified to exit program)
Get user input
Write user input to pipe 1 to send to output function
Check what the user entered
    If ctrl K the kill all processes and exit
    If ‘T’ then write buffered user data to pipe 2 to translate function
        Clear the buffer
        Set buffer counter to 0
        Set kill program counter
        Enable Linux terminal settings
    If ‘E’ then write buffered user data to pipe 2 to translate function
        Clear buffer
        Set buffer counter to 0
    If ‘K’ then clear buffered user data
        Send buffer down pipe 2 to translate function
    Default
        Add user inputted data to buffer
        Increment buffer counter
```

Output function

```
Close writing to pipe 1 and 3
Go into infinite loop
If data received from pipe1 is an ‘E’, ‘K’, or ‘T’ then read from pipe 3
    Print contents
Else read from pipe 1
Print content
```

Translate

```
Close writing to pipe 2
Close reading from pipe 3
Go into infinite loop
If there is data to read from pipe 2 then start iterating thru data received (use two counters, ‘i’ and ‘j’. ‘i’ is used to iterate thru data received. ‘j’ is used to iterate thru temp buffer that will be copying data from data received)
    Check each character being iterated thru
        If ‘X’ then decrement ‘j’ by 2
    If ‘a’ then make temp buffer data element equal ‘z’
    Default
        Copy data element from data received into temp buffer
If ‘j’is equal to 0
    Clear the temp buffer
    Make the first element equal a character space (‘   ‘)
    Set ‘j’ to 1
Write data from temp buffer to pipe 3 to go to output function
Clear temp buffer
Clear data received buffer
```