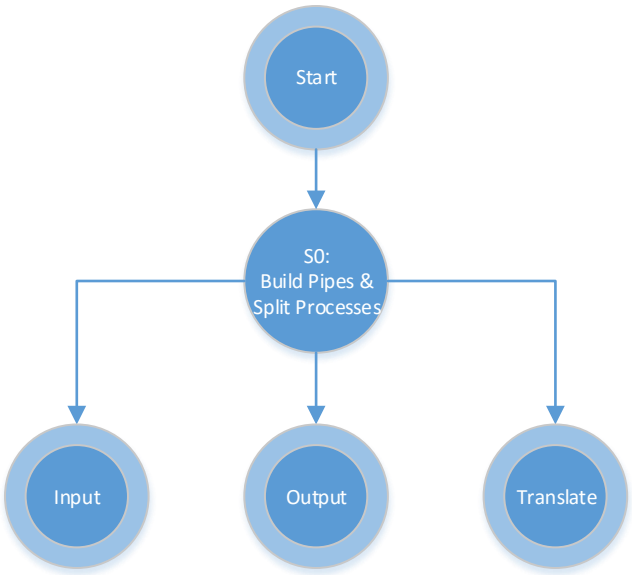
Carson Roscoe January 10th, 2016

COMP4981 Design Work

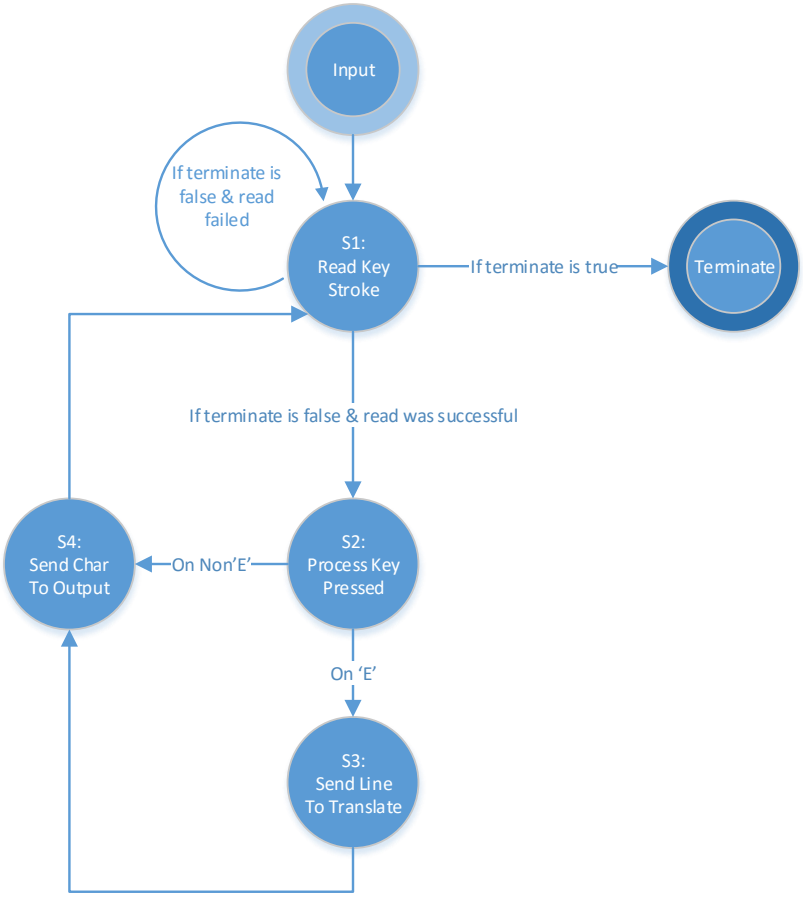
Contents

State Diagrams

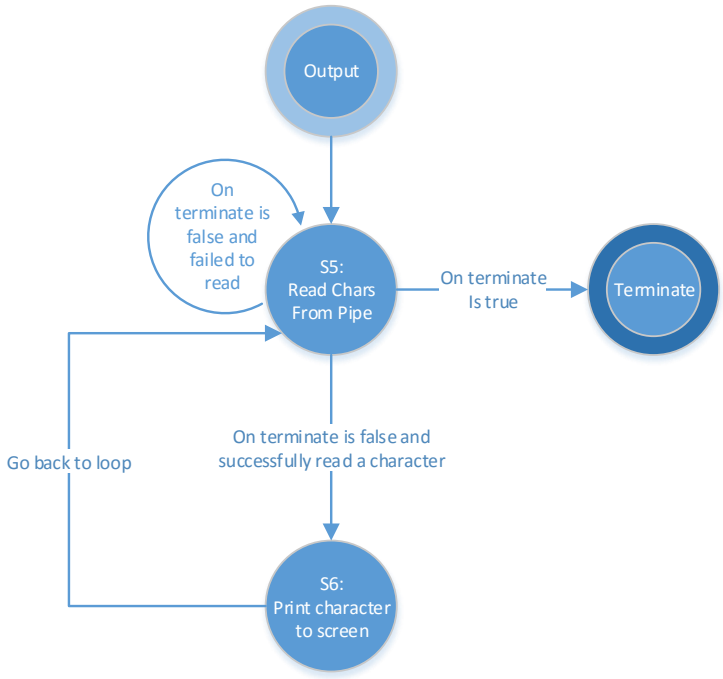
SHARED STATES



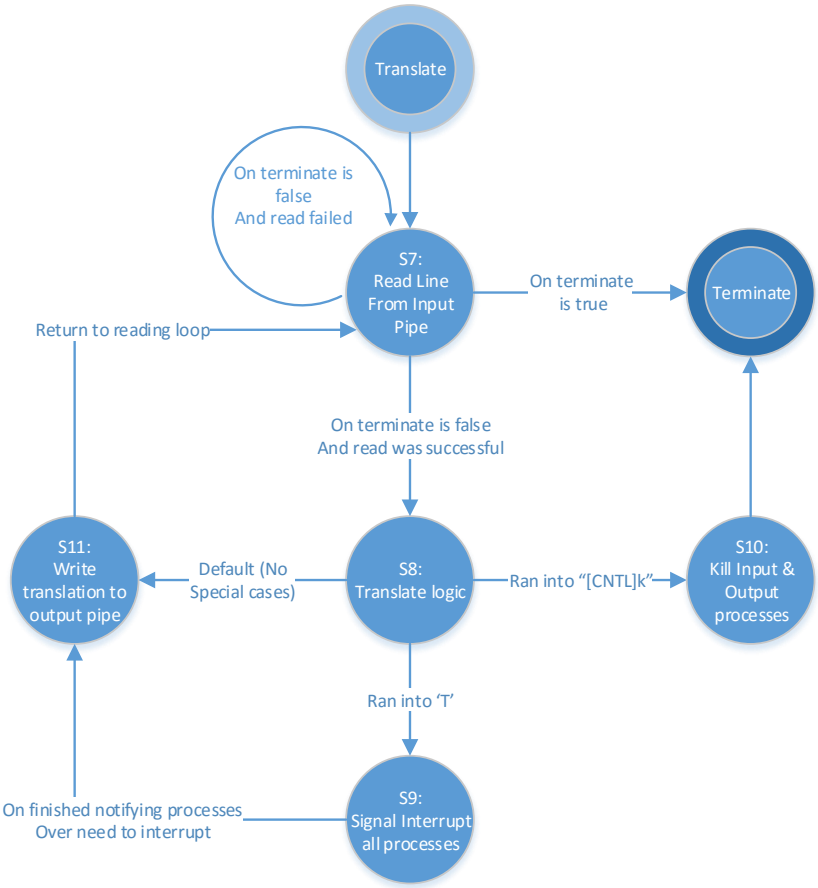
INPUT PROCESS' STATES



OUTPUT PROCESS' STATES



TRANSLATES PROCESS' STATES



Pseudocode

SHARED STATES

Method terminate:

```
//Used to modify the processes 'terminate' boolean to equal true.
//This allows for a natural exit of all process' infinite loops.
terminate = true
```

S0: Build Pipes & Split Processes:

```
boolean terminate defaulted to false.
int array pipeOutput size 2
int array pids size 3
int output, input
Make signal interrupt(SIGINT) invoke Method terminate
Disable terminals ability to echo, process I/O and ignore carriage return
Fork and store result into output
If current process is the child of the fork
        GOTO OUTPUT's S#
else
        int array pipeTranslate size 2
        Fork and store result into input
        If current process is the child of the fork
                GOTO INPUTS's S#
        else
                put input, output and current processes PID into array pids
                GOTO TRANSLATE's S#
```

INPUT PROCESS' STATES

S1: Read Key Stroke:

```
create a buffer of chars
while terminate is false
        Read the next character pressed by the keyboard and store it into a variable
        If read was successful
                GOTO INPUT's S2
exit process
```

S2: Key Pressed:

If key pressed was E
    GOTO S3
Else
IF key pressed was K
    Clear buffer
    GOTO S4
Else
    add key pressed to buffer
    GOTO S4

S3: Send Line To Transfer

Write all chars in buffer to translate's pipe.
GOTO S4

S4: Send Char To Output

Write key pressed to output's pipe
GOTO S1

OUTPUT PROCESS' STATES

S5: Read Chars From Pipe

while terminate is false
    Read the next character from the pipe
        GOTO S6
exit process

S6: Print Character To Screen

Print character read from pipe to screen
GOTO S5

TRANSLATE PROCESS' STATES

S7: Read Line From Input Pipe

While terminate is false
    Read everything from input pipe and store in a buffer
    If read was successful
        GOTO S8
return to terminal the ability to process I/O, echo and carriage return.
exit process

S8: Translate Logic

```
for each character in buffer
        if character is an 'X'
                erase last character drawn onto the screen
        else if character is 'a'
                change 'a' to 'z'
                GOTO S11
        else if character is 'T'
                GOTO S9
        else if character is ctrl-k
                GOTO S10
        else (default)
                GOTO S11
```

S9: Signal Interrupt All Processes

```
for each process ID in pids
        signal interrupt that process by its ID
GOTO S11
```

S10: Kill Input & Output processes

```
for each process ID in pids
        signal kill that process by its ID
exit process
```

S11: Write translation to output pipe

```
write buffer to output pipe
GOTO S7
```