Desired Output For Project 8

BasicSerialTest

```
% blitz -g -wait os -raw
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  BasicSerialTest  ==========

This test should be run in raw mode.

Hit the "a" key.  Do not hit ENTER or RETURN...
Returned value is correct.
The buffer was updated correctly.

Please type "abc".  Do not hit ENTER or RETURN...
Returned value is correct.
The buffer was updated correctly.

Please type "hello".  Do not hit ENTER or RETURN...
Returned value is correct.
Returned value is correct.
The buffer was updated correctly.

Please type control-J.  Do not hit ENTER or RETURN...
Returned value is correct.
The buffer was updated correctly.

Please type control-M.  Do not hit ENTER or RETURN...
Returned value is correct.
The buffer was updated correctly.

Please type control-H.  Do not hit ENTER or RETURN...
Returned value is correct.
The buffer was updated correctly.

Please type control-D.  Do not hit ENTER or RETURN...
Returned value is correct.
The buffer was not modified, as expected.
```

```
==========  Test Complete  ==========
```

KeyTest (in raw mode)

```
% blitz -g -wait os -raw
Beginning execution...
====================  KPL PROGRAM STARTING  ====================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  KeyTest  ==========

This test waits for a single character and then prints the value of that character.

  1. Start by typing "abcABC123"
  2. See what happens with characters like cntl-H (Backspace), cntl-J (NL) and
cntl-M (CR).
  3. See what happens when you hit keys labelled SPACE, ENTER/RETURN, TAB,
DEL/BACKSPACE, ARROW KEYS and ESC.
  4. See what happens when you hit cntl-D (EOF).
  5. Try this test in cooked mode and in raw mode.

This test will terminate when 'q' is typed.

(To change to cooked mode, type control-C, "cooked", and "g" to resume execution.
To change to raw mode, type control-C, ENTER, ENTER, "raw", and "g" to resume
execution.)

ch = 0x00000061 (decimal 97)
ch = 0x00000062 (decimal 98)
ch = 0x00000063 (decimal 99)
ch = 0x00000041 (decimal 65)
ch = 0x00000042 (decimal 66)
ch = 0x00000043 (decimal 67)
ch = 0x00000031 (decimal 49)
ch = 0x00000032 (decimal 50)
ch = 0x00000033 (decimal 51)
ch = 0x00000008 (decimal 8)
ch = 0x0000000A (decimal 10)
ch = 0x0000000A (decimal 10)
ch = 0x00000020 (decimal 32)
ch = 0x0000000A (decimal 10)
ch = 0x00000009 (decimal 9)
ch = 0x0000007F (decimal 127)
ch = 0x0000001B (decimal 27)
ch = 0x0000005B (decimal 91)
ch = 0x00000044 (decimal 68)
ch = 0x0000001B (decimal 27)
ch = 0x0000005B (decimal 91)
ch = 0x00000043 (decimal 67)
ch = 0x0000001B (decimal 27)

*****  WARNING: Returned value from Read is zero; This should only occur when
control-D is typed
ch = 0x00000061 (decimal 97)
ch = 0x00000062 (decimal 98)
ch = 0x00000063 (decimal 99)
ch = 0x00000071 (decimal 113)

==========  Test Complete  ==========
```

KeyTest (in cooked mode)

```
% blitz -g -wait os
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  KeyTest  ==========

This test waits for a single character and then prints the value of that character.

  1. Start by typing "abcABC123"
  2. See what happens with characters like cntl-H (Backspace), cntl-J (NL) and
cntl-M (CR).
  3. See what happens when you hit keys labelled SPACE, ENTER/RETURN, TAB,
DEL/BACKSPACE, ARROW KEYS and ESC.
  4. See what happens when you hit cntl-D (EOF).
  5. Try this test in cooked mode and in raw mode.

This test will terminate when 'q' is typed.

(To change to cooked mode, type control-C, "cooked", and "g" to resume execution.
To change to raw mode, type control-C, ENTER, ENTER, "raw", and "g" to resume
execution.)


*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
abcABC123^H^H^H
ch = 0x00000061 (decimal 97)
ch = 0x00000062 (decimal 98)
ch = 0x00000063 (decimal 99)
ch = 0x00000041 (decimal 65)
ch = 0x00000042 (decimal 66)
ch = 0x00000043 (decimal 67)
ch = 0x00000031 (decimal 49)
ch = 0x00000032 (decimal 50)
ch = 0x00000033 (decimal 51)
ch = 0x00000008 (decimal 8)
ch = 0x00000008 (decimal 8)
ch = 0x00000008 (decimal 8)
ch = 0x0000000A (decimal 10)

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
a b      d
ch = 0x00000061 (decimal 97)
ch = 0x00000020 (decimal 32)
ch = 0x00000062 (decimal 98)
ch = 0x00000009 (decimal 9)
ch = 0x00000020 (decimal 32)
ch = 0x00000064 (decimal 100)
ch = 0x0000000A (decimal 10)

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
arrow^[[C^[[A^[[B^[[Descape^[EOF^D
ch = 0x00000061 (decimal 97)
ch = 0x00000072 (decimal 114)
ch = 0x00000072 (decimal 114)
ch = 0x0000006F (decimal 111)
ch = 0x00000077 (decimal 119)
ch = 0x0000001B (decimal 27)
ch = 0x0000005B (decimal 91)
ch = 0x00000043 (decimal 67)
ch = 0x0000001B (decimal 27)
ch = 0x0000005B (decimal 91)
ch = 0x00000041 (decimal 65)
ch = 0x0000001B (decimal 27)
ch = 0x0000005B (decimal 91)
ch = 0x00000042 (decimal 66)
ch = 0x0000001B (decimal 27)
ch = 0x0000005B (decimal 91)
ch = 0x00000044 (decimal 68)
ch = 0x00000065 (decimal 101)
ch = 0x00000073 (decimal 115)
ch = 0x00000063 (decimal 99)
ch = 0x00000061 (decimal 97)
ch = 0x00000070 (decimal 112)
ch = 0x00000065 (decimal 101)
ch = 0x0000001B (decimal 27)
ch = 0x00000045 (decimal 69)
ch = 0x0000004F (decimal 79)
ch = 0x00000046 (decimal 70)
ch = 0x0000000A (decimal 10)

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
```

```
q
ch = 0x00000071 (decimal 113)

==========  Test Complete  ==========
```

EchoTest

```
% blitz -g -wait os
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  EchoTest  ==========

This test reads characters from the terminal.  It echoes each character, as it is
received.

  1. Start by typing "abc\n"
  2. See what happens with characters like cntl-H (Backspace), cntl-J (NL) and
cntl-M (CR).
  3. See what happens when you hit keys labelled SPACE, ENTER/RETURN, TAB,
DEL/BACKSPACE, and ESC.
  4. See what happens when you hit cntl-D (EOF).
  5. Try this test in cooked mode and in raw mode.
  6. While in raw mode, see what happens with sequences from page 342 in the
textbook.
     For example, try typing these sequences:
         ESC  [  7  m
         ESC  [  5  A
         control-g
         up-arrow

This test will terminate when 'q' is typed.

(To change to cooked mode, type control-C, "cooked", and "g" to resume execution.
To change to raw mode, type control-C, ENTER, ENTER, "raw", and "g" to resume
execution.)


*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
abc
abc
```

```
*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
hello there^H^H^H^H^Hall
hello allre

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
hello all
hello all

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
^D
*****  EOF on input ignored: Use Control-C to halt execution  *****

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
^D
*****  EOF on input ignored: Use Control-C to halt execution  *****

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
^[[7m

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
^G***  Execution suspended on 'wait' instruction; waiting for additional user input
*****
^[[5A

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
^[[A

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
q

==========  Test Complete  ==========
```

LineEchoTest (in cooked mode)

```
% blitz -g -wait os
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
```

```
Loading initial program...

==========  LineEchoTest  ==========

This program reads an entire line of input (up to 30 characters) into a buffer.
After this program gets the entire line, it prints it.  Since this program
does not echo characters as typed, you will not see the characters until after
the entire line is completed by typing ENTER, when running the emulator in raw
mode.  In cooked mode, the host (Unix) will echo characters and then, after the
line is complete, this program will (re) print the line.

  1. Start by typing "abc\n"
  2. See what happens with characters like cntl-H (Backspace), cntl-J (NL) and
cntl-M (CR).
  3. See what happens when you hit keys labelled SPACE, ENTER/RETURN, TAB,
DEL/BACKSPACE, and ESC.
  4. See what happens when you hit cntl-D (EOF).
  5. See what happens when the size of the buffer is exceeded.
  6. In cooked mode, see how the host (Unix) processes editing keys, such as
backspace.

This test will terminate when the first character entered is 'q'.

(To change to cooked mode, type control-C, "cooked", and "g" to resume execution.
To change to raw mode, type control-C, ENTER, ENTER, "raw", and "g" to resume
execution.)


*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
abc

Number of characters entered = 4
abc

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****

abc

Number of characters entered = 1


Number of characters entered = 4
abc

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
hello there^H^H^H

Number of characters entered = 15
hello there

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
hello

Number of characters entered = 6
```

```
hello

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Number of characters entered = 30
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Number of characters entered = 30
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Number of characters entered = 6
aaaaa

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
q

Number of characters entered = 2
q

==========  Test Complete  ==========
```

LineEchoTest (in raw mode)

```
% blitz -g -wait os -raw
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  LineEchoTest  ==========

This program reads an entire line of input (up to 30 characters) into a buffer.
After this program gets the entire line, it prints it.  Since this program
does not echo characters as typed, you will not see the characters until after
the entire line is completed by typing ENTER, when running the emulator in raw
mode.  In cooked mode, the host (Unix) will echo characters and then, after the
line is complete, this program will (re) print the line.

  1. Start by typing "abc\n"
  2. See what happens with characters like cntl-H (Backspace), cntl-J (NL) and
cntl-M (CR).
  3. See what happens when you hit keys labelled SPACE, ENTER/RETURN, TAB,
DEL/BACKSPACE, and ESC.
  4. See what happens when you hit cntl-D (EOF).
  5. See what happens when the size of the buffer is exceeded.
  6. In cooked mode, see how the host (Unix) processes editing keys, such as
backspace.

This test will terminate when the first character entered is 'q'.

(To change to cooked mode, type control-C, "cooked", and "g" to resume execution.
To change to raw mode, type control-C, ENTER, ENTER, "raw", and "g" to resume
execution.)


Number of characters entered = 6
abcde

Number of characters entered = 17
hello there

Number of characters entered = 2
q

==========  Test Complete  ==========
```

EOFTest

```
blitz -g -wait os -raw
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  EOFTest  ==========

This test should be run in 'raw' mode.

This function tests the handling of control-D.  Control-D is the
enf-of-file character.  When typed, it should cause an immediate
return from the Read syscall.  If no other characters have been
typed first, then the count returned from Read will be zero, which
many programs will interpret as 'end-of-file'.

Please hit control-D next.  The ENTER key should not be necessary...
Okay.
Please type "abc" followed by control-D.  The ENTER key should not be necessary...
Okay.

==========  Test Complete  ==========
```

OpenCloseTest

```
% blitz -g -wait os -raw
Beginning execution...
====================  KPL PROGRAM STARTING  ===================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  OpenCloseTerminalTest  ==========

Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Closing all 10 fileDescriptors...
Opening 'terminal' 10 times...
Attempting to open 'terminal' one more time, which should fail.
The syscall correctly returns -1.
Closing all 10 fileDescriptors...

==========  Test Complete  ==========
```

TerminalErrorTest

```
% blitz -g -wait os
Beginning execution...
```

```
===================  KPL PROGRAM STARTING  ==================
Initializing Thread Scheduler...
Initializing Process Manager...
Initializing Thread Manager...
Initializing Frame Manager...
AllocateRandomFrames called.  NUMBER_OF_PHYSICAL_PAGE_FRAMES = 512
Initializing Disk Driver...
Initializing Serial Driver...
Initializing File Manager...
Serial handler thread running...
Loading initial program...

==========  TerminalErrorTest  ==========

(This test should be run in cooked mode.)
Opening 'terminal' file...
Reading with negative sizeInBytes...
Okay.
Reading with negative sizeInBytes...
Okay.
Reading with a pointer to a page which is read-only, which should be an error...
Please type "abc\n" next.

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
abc
Okay.
Reading with a pointer which isn't in our address space, which should be an
error...
Please type "xyz\n" next.

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
xyz
Okay.
Reading with a pointer which is near the end of our address space...
Please type "123456\n" next.

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
123456
Okay.
Reading with a pointer that crosses a page boundary...
Please type "abcdef\n" next.

*****  Execution suspended on 'wait' instruction; waiting for additional user input
*****
abcdef
Okay.
Writing with negative sizeInBytes...
Okay.
Writing with negative sizeInBytes...
Okay.
Writing with a pointer that crosses a page boundary...


==== This should print "GREETINGS" next ====
                        GREETINGS
```

```
Okay.
Writing with a pointer to a page which is read-only, which should be okay...


==== This should print "KERNEL CODE" next ====
                        KERNEL CODE

Writing with a pointer which isn't in our address space, which should be an
error...
Okay.

==========  Test Complete  ==========
```