

CS 510: Foundations of Operating Systems  
Fall 2017, Dr. Jonathan Walpole

Stephen Braich  
sbraich@pdx.edu  
PSU ID: 953378420

## **Programming Project 1:**

# Introduction to the Blitz Tools

October 2<sup>nd</sup>, 2017

```
steve@ubuntu14large: ~
sbraich@ada:~/cs510os/p1$ asm Hello.s
sbraich@ada:~/cs510os/p1$ ldd Hello.o -o Hello
sbraich@ada:~/cs510os/p1$ blitz -g Hello
Beginning execution...
Hello, world!

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
000080: A1FFFFB8      jmp      0xFFFFB8      ! targetAddr = main

Entering machine-level debugger...
=====
=====
===== The BLITZ Machine Emulator =====
=====
===== Copyright 2001-2007, Harry H. Porter III =====
=====
=====

Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> q
Number of Disk Reads      = 0
Number of Disk Writes     = 0
Instructions Executed      = 1705
Time Spent Sleeping       = 0
Total Elapsed Time        = 1705
sbraich@ada:~/cs510os/p1$ asm Echo.s
sbraich@ada:~/cs510os/p1$ ldd Echo.o -o Echo
sbraich@ada:~/cs510os/p1$ blitz Echo
=====
=====
===== The BLITZ Machine Emulator =====
=====
===== Copyright 2001-2007, Harry H. Porter III =====
=====
=====

Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> g
Beginning execution...
abcd
abcd
this is a test
this is a test
q
q
**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
cont:
0000A4: A1FFFFAC      jmp      0xFFFFAC      ! targetAddr = loop
>
```

```
steve@ubuntu14large: ~
sbraich@ada:~/cs510os/p1$ vi HelloWorld.c
sbraich@ada:~/cs510os/p1$ kpl -unsafe System
sbraich@ada:~/cs510os/p1$ asm System.s
sbraich@ada:~/cs510os/p1$ lpl HelloWorld
-bash: lpl: command not found
sbraich@ada:~/cs510os/p1$ asm Runtime.s
sbraich@ada:~/cs510os/p1$ ldd Runtime.o System.o HelloWorld.o -o HelloWorld
sbraich@ada:~/cs510os/p1$ blitz -g HelloWorld
Beginning execution...
===== KPL PROGRAM STARTING =====
Hello, world...

===== KPL PROGRAM TERMINATION =====

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
000D98: C0100000      sethi    0x0000,r1      ! 0x00000DA8 = 3496 (noGoMessage)

Entering machine-level debugger...
=====
===== The BLITZ Machine Emulator =====
===== Copyright 2001-2007, Harry H. Porter III =====
=====

Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> q
Number of Disk Reads      = 0
Number of Disk Writes     = 0
Instructions Executed      = 945
Time Spent Sleeping       = 0
    Total Elapsed Time    = 945
sbraich@ada:~/cs510os/p1$ make
kpl HelloWorld
asm HelloWorld.s
ldd Runtime.o System.o HelloWorld.o -o HelloWorld
sbraich@ada:~/cs510os/p1$ blitz -g HelloWorld
Beginning execution...
===== KPL PROGRAM STARTING =====
Hello, world...
The value of b is 12

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
0028A4: 8B1EFFF0      load    [r14+0xFFF0],r1 ! decimal: -16

Entering machine-level debugger...
=====
===== The BLITZ Machine Emulator =====
```

```
steve@ubuntu14large: ~  
  
Entering machine-level debugger...  
=====
```

```
=====
The BLITZ Machine Emulator
=====
Copyright 2001-2007, Harry H. Porter III
=====
=====
```

```
Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> g
Beginning execution...
The value of b is 13

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
0028A4: 8B1EFFF0      load    [r14+0xFFFF0],r1 ! decimal: -16
> q
Number of Disk Reads    = 0
Number of Disk Writes   = 0
Instructions Executed    = 694
Time Spent Sleeping     = 0
    Total Elapsed Time  = 694
sbraich@ada:~/cs510os/p1$ blitz -g HelloWorld
Beginning execution...
===== KPL PROGRAM STARTING =====
Hello, world...
The value of b is 12

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
0028A4: 8B1EFFF0      load    [r14+0xFFFF0],r1 ! decimal: -16

Entering machine-level debugger...
=====
```

```
=====
The BLITZ Machine Emulator
=====
Copyright 2001-2007, Harry H. Porter III
=====
=====
```

```
Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> quit
Number of Disk Reads    = 0
Number of Disk Writes   = 0
Instructions Executed    = 609
Time Spent Sleeping     = 0
    Total Elapsed Time  = 609
sbraich@ada:~/cs510os/p1$ blitz -g HelloWorld
```



```
steve@ubuntu14large: ~
Beginning execution...
===== KPL PROGRAM STARTING =====
Hello, world...
The value of b is 12

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
0028A4: 8B1EFFF0      load    [r14+0xFFFF0],r1 ! decimal: -16

Entering machine-level debugger...
=====
===== The BLITZ Machine Emulator =====
===== Copyright 2001-2007, Harry H. Porter III =====
=====
Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> quit
Number of Disk Reads      = 0
Number of Disk Writes     = 0
Instructions Executed      = 609
Time Spent Sleeping       = 0
Total Elapsed Time        = 609
sbraich@ada:~/cs510os/p1$ blitz -g HelloWorld
Beginning execution...
===== KPL PROGRAM STARTING =====
Hello, world...
The value of b is 12

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
0028A4: 8B1EFFF0      load    [r14+0xFFFF0],r1 ! decimal: -16

Entering machine-level debugger...
=====
===== The BLITZ Machine Emulator =====
===== Copyright 2001-2007, Harry H. Porter III =====
=====
Enter a command at the prompt. Type 'quit' to exit or 'help' for
info about commands.
> help
=====
This program accepts commands typed into the terminal. Each command
should be typed without any arguments; the commands will prompt for
arguments when needed. Case is not significant. Some abbreviations
are allowed, as shown. Typing control-C will halt execution.
```

steve@ubuntu14large: ~

The available commands are:

```
quit      - Terminate this program
q
help      - Produce this display
h
info      - Display the current state of the machine
i
dumpMem   - Display the contents of memory
dm
setmem    - Used to alter memory contents
fmem      - Display floating point values from memory
go        - Begin or resume BLITZ instruction execution
g
step      - Single step; execute one machine-level instruction
s
t         - Single step; execute one KPL statement
u         - Execute continuously until next KPL call, send, or return statement
stepn     - Execute N machine-level instructions
r         - Display all the integer registers
r1        - Change the value of register r1
...
r15       - Change the value of register r15
float     - Display all the floating-point registers
f
f0        - Change the value of floating-point register f0
...
f15       - Change the value of floating-point register f15
dis       - Disassemble several instructions
d         - Disassemble several instructions from the current location
hex       - Convert a user-entered hex number into decimal and ascii
dec       - Convert a user-entered decimal number into hex and ascii
ascii     - Convert a user-entered ascii char into hex and decimal
setI      - Set the I bit in the Status Register
setS      - Set the S bit in the Status Register
setP      - Set the P bit in the Status Register
setZ      - Set the Z bit in the Status Register
setV      - Set the V bit in the Status Register
setN      - Set the N bit in the Status Register
clearI    - Clear the I bit in the Status Register
clearS    - Clear the S bit in the Status Register
clearP    - Clear the P bit in the Status Register
clearZ    - Clear the Z bit in the Status Register
clearV    - Clear the V bit in the Status Register
clearN    - Clear the N bit in the Status Register
setPC     - Set the Program Counter (PC)
setPTBR   - Set the Page Table Base Register (PTBR)
setPTLR   - Set the Page Table Length Register (PTLR)
pt        - Display the Page Table
trans     - Perform page table translation on a single address
cancel    - Cancel all pending interrupts
labels    - Display the label table
find      - Find a label by name
```



```
steve@ubuntu14large: ~
find      - Find a label by name
find2     - Find a label by value
add       - Add a new label, inserting it into the indexes
reset     - Reset the machine state and re-read the a.out file
io        - Display the state of the I/O devices
read      - Read a word from memory-mapped I/O region
write     - Write a word to memory-mapped I/O region
raw       - Switch serial input to raw mode
cooked    - Switch serial input to cooked mode
input     - Enter input characters for future serial I/O input
format    - Create and format a BLITZ disk file
sim       - Display the current simulation constants
stack     - Display the KPL calling stack
st
frame     - Display the current activation frame
fr
up        - Move up in the activation frame stack
down      - Move down in the activation frame stack

=====
> go
Beginning execution...
The value of b is 13

**** A 'debug' instruction was encountered ****
Done! The next instruction to execute will be:
0028A4: 8B1EFFF0      load    [r14+0xFFFF],r1 ! decimal: -16
> step
Done! The next instruction to execute will be:
0028A8: 8F1F0000      store   r1,[r15+0x0000] ! decimal: 0 (PowerOnReset)
> t
About to execute FUNCTION CALL                in bar (HelloWorld.c, line 21)
    time = 698
> reset
Resetting all CPU registers and re-reading file "HelloWorld"...
> info
=====
Memory size = 0x01000000      ( decimal: 16777216      )
Page size   = 0x00002000      ( decimal: 8192        )
.text Segment
    addr     = 0x00000000      ( decimal: 0          )
    size     = 0x00004000      ( decimal: 16384       )
.data Segment
    addr     = 0x00004000      ( decimal: 16384       )
    size     = 0x00006000      ( decimal: 24576       )
.bss Segment
    addr     = 0x0000A000      ( decimal: 40960       )
    size     = 0x00000000      ( decimal: 0           )
===== USER REGISTERS =====
r0 = 0x00000000      ( decimal: 0 )
r1 = 0x00000000      ( decimal: 0 )
r2 = 0x00000000      ( decimal: 0 )
r3 = 0x00000000      ( decimal: 0 )
```

```
steve@ubuntu14large: ~  
===== USER REGISTERS =====  
r0 = 0x00000000 ( decimal: 0 )  
r1 = 0x00000000 ( decimal: 0 )  
r2 = 0x00000000 ( decimal: 0 )  
r3 = 0x00000000 ( decimal: 0 )  
r4 = 0x00000000 ( decimal: 0 )  
r5 = 0x00000000 ( decimal: 0 )  
r6 = 0x00000000 ( decimal: 0 )  
r7 = 0x00000000 ( decimal: 0 )  
r8 = 0x00000000 ( decimal: 0 )  
r9 = 0x00000000 ( decimal: 0 )  
r10 = 0x00000000 ( decimal: 0 )  
r11 = 0x00000000 ( decimal: 0 )  
r12 = 0x00000000 ( decimal: 0 )  
r13 = 0x00000000 ( decimal: 0 )  
r14 = 0x00000000 ( decimal: 0 )  
r15 = 0x00000000 ( decimal: 0 )  
===== SYSTEM REGISTERS =====  
r0 = 0x00000000 ( decimal: 0 )  
r1 = 0x00000000 ( decimal: 0 )  
r2 = 0x00000000 ( decimal: 0 )  
r3 = 0x00000000 ( decimal: 0 )  
r4 = 0x00000000 ( decimal: 0 )  
r5 = 0x00000000 ( decimal: 0 )  
r6 = 0x00000000 ( decimal: 0 )  
r7 = 0x00000000 ( decimal: 0 )  
r8 = 0x00000000 ( decimal: 0 )  
r9 = 0x00000000 ( decimal: 0 )  
r10 = 0x00000000 ( decimal: 0 )  
r11 = 0x00000000 ( decimal: 0 )  
r12 = 0x00000000 ( decimal: 0 )  
r13 = 0x00000000 ( decimal: 0 )  
r14 = 0x00000000 ( decimal: 0 )  
r15 = 0x00000000 ( decimal: 0 )  
===== FLOATING-POINT REGISTERS =====  
f0 = 0x00000000 00000000 ( value = 0 )  
f1 = 0x00000000 00000000 ( value = 0 )  
f2 = 0x00000000 00000000 ( value = 0 )  
f3 = 0x00000000 00000000 ( value = 0 )  
f4 = 0x00000000 00000000 ( value = 0 )  
f5 = 0x00000000 00000000 ( value = 0 )  
f6 = 0x00000000 00000000 ( value = 0 )  
f7 = 0x00000000 00000000 ( value = 0 )  
f8 = 0x00000000 00000000 ( value = 0 )  
f9 = 0x00000000 00000000 ( value = 0 )  
f10 = 0x00000000 00000000 ( value = 0 )  
f11 = 0x00000000 00000000 ( value = 0 )  
f12 = 0x00000000 00000000 ( value = 0 )  
f13 = 0x00000000 00000000 ( value = 0 )  
f14 = 0x00000000 00000000 ( value = 0 )  
f15 = 0x00000000 00000000 ( value = 0 )  
===== PC =====  
PC = 0x00000000 ( decimal: 0 )
```



```

steve@ubuntu14large: ~
f15 = 0x00000000 00000000 ( value = 0 )
=====
PC = 0x00000000 ( decimal: 0 )
PTBR = 0x00000000 ( decimal: 0 )
PTLR = 0x00000000 ( decimal: 0 )
-----
SR = 0x00000010 = 0000 0000 0000 0000 0000 0000 0001 0000 --IS PZVN
    I = 0   Interrupts Disabled
    S = 1   System Mode
    P = 0   Paging Disabled
    Z = 0   Not Zero
    V = 0   No Overflow
    N = 0   Not Negative
=====
Pending Interrupts          = 0x00000002
TIMER_INTERRUPT
System Trap Number          = 0x00000000
Page Invalid Offending Address = 0x00000000
Page Readonly Offending Address = 0x00000000
Time of next timer event     = 5005
Time of next disk event      = 2147483647
Time of next serial in event = 0
Time of next serial out event = 2147483647
Current Time                 = 0
Time of next event           = 0
Time Spent Sleeping          = 0
Instructions Executed         = 0
Number of Disk Reads          = 0
Number of Disk Writes         = 0
=====
The next instruction to execute will be:
    PowerOnReset:
    _entry:
000000: A1000CD8      jmp      0x000CD8      ! targetAddr = RuntimeStartup
About to execute ***INVALID HIGH-LEVEL STATEMENT CODE IN REGISTER r10***
> stack
    Function/Method          Frame Addr   Execution at...
    =====
Bottom of activation frame stack!
> frame
===== Frame number 0 (where StackTop = 0) =====
Bottom of activation frame stack!
Resetting current frame to top of stack!
> up
Already at top of stack!
===== Frame number 0 (where StackTop = 0) =====
Bottom of activation frame stack!
Resetting current frame to top of stack!
> down
===== Frame number 1 (where StackTop = 0) =====
Invalid activation frame stack! The frame pointer is 0x00000000.
Resetting current frame to top of stack!
>

```