Lab 1

Step 7:Assemble, Link, and Execute the “Hello” Program

Type :

```
asm Hello.s #Assemble
```

Then we will create a new file named "Hello.o", which is named "Assemble".

Then Type:

```
lddd Hello.o -o Hello #Link
```

As the pdf said, although the program **Hello.s** is stand-alone, which means it does not need any library functions and does not rely on any operating system, it has to be linked to produce an executable.

"-o Hello" rename the executable from "a.out" to "Hello".

Last Type:

```
blitz –g Hello #Execute
```

output:

```
harryovo@harryovo-virtual-machine: ~/Desktop/lab1
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ ls
DISK    Echo.s   Hello.s       HelloWorld.h  makefile   Runtime.s  System.o
Echo    Hello    HelloWorld    HelloWorld.o  proj1.pdf  System.c   System.s
Echo.o  Hello.o  HelloWorld.c  HelloWorld.s  Runtime.o  System.h
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm Hello.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ lddd Hello.o -o Hello
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ 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
harryovo@harryovo-virtual-machine:~/Desktop/lab1$
```

Step 8:Run the “Echo” Program

Type in the following commands:

```
asm Echo.s
lddd Echo.o –o Echo
blitz Echo
```

then we need to type in "g" to run.

output:

```
harryovo@harryovo-virtual-machine: ~/Desktop/lab1
Time Spent Sleeping     = 0
    Total Elapsed Time  = 1705
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm Echo.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ lddd Echo.o -o Echo
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ 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
> g
Beginning execution...

try again
try again
try to type with a q
try to type with a q
****  A 'debug' instruction was encountered  *****
Done!  The next instruction to execute will be:
                    cont:
0000A4: A1FFFFAC       jmp     0xFFFFAC         ! targetAddr = loop
> q
Number of Disk Reads    = 0
Number of Disk Writes   = 0
Instructions Executed   = 1512859460
Time Spent Sleeping     = 0
    Total Elapsed Time  = 1512859460
```

Step 9:Compile and Execute a KPL Program called “HelloWorld”

Type the following commands:

```
kpl -unsafe System
asm System.s
kpl HelloWorld
asm HelloWorld.s
asm Runtime.s
lddd Runtime.o System.o HelloWorld.o -o HelloWorld
```

output:

```
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ kpl -unsafe System
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm System.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ kpl HelloWorld
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm HelloWorld.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm Runtime.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ lddd Runtime.o System.o HelloWorld.o -o HelloWorld
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ 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
```

Step 10:Modify the HelloWorld Program

Modify the HelloWorld.c program by un-commenting the line "--foo (10)"

output:

```
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ kpl -unsafe System
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm System.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ kpl HelloWorld
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm HelloWorld.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ asm Runtime.s
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ lddd Runtime.o System.o HelloWorld.o -o HelloWorld
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ 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        =====
=====                                           =====
=====  Copyright 2001-2007, Harry H. Porter III  =====
=====                                           =====
=====================================================

Enter a command at the prompt.  Type 'quit' to exit or 'help' for
info about commands.
>
```

there is an extra line "The value of b is 12".

Step 11: Try Some of the Emulator Commands

1. t (program will execute after "debug" sign)

which means executing a single high-level KPL language statement at a time. Typing “t” several times to walk through the execution of the HelloWorld program.

```
harryovo@harryovo-virtual-machine: ~/Desktop/lab1
harryovo@harryovo-virtual-machine:~/Desktop/lab1$ 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        =====
=====                                           =====
=====  Copyright 2001-2007, Harry H. Porter III  =====
=====                                           =====
=====================================================

Enter a command at the prompt.  Type 'quit' to exit or 'help' for
info about commands.
> t
About to execute FUNCTION CALL                     in bar (HelloWorld.c, line 21)  time = 613
> t
About to execute FUNCTION ENTRY                    in bar (HelloWorld.c, line 14)  time = 632
> t
About to execute ASSIGN statement                  in bar (HelloWorld.c, line 16)  time = 634
> t
About to execute FUNCTION CALL (external function) in bar (HelloWorld.c, line 17)  time = 646
> t
The value of b is 13About to execute FUNCTION CALL                     in bar (HelloWorld.c, line 19)  time = 663
> t
About to execute FUNCTION ENTRY                    in nl (System.c, line 48)  time = 676
> t
About to execute FUNCTION CALL (external function) in nl (System.c, line 49)  time = 680
> t

About to execute RETURN statement                  in nl (System.c, line 49)  time = 687
> t
About to execute DEBUG statement                   in bar (HelloWorld.c, line 20)  time = 693
> t

****  A 'debug' instruction was encountered  *****
```

2. i (display details of the cpu)

output:

```
About to execute FUNCTION ENTRY                    in nl (System.c, line 48)  time = 676
> i
============================
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 )
  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  = 0x0000000D     ( decimal: 13          )
  r3  = 0x00000012     ( decimal: 18          )
  r4  = 0x8CC97375     ( decimal: -1932954763 )
  r5  = 0x00000000     ( decimal: 0 )
  r6  = 0x00000000     ( decimal: 0 )
  r7  = 0x00000000     ( decimal: 0 )
  r8  = 0x00000000     ( decimal: 0 )
  r9  = 0x00000000     ( decimal: 0 )
  r10 = 0x00004655     ( decimal: 18005       )
  r11 = 0x00000000     ( decimal: 0 )
```

```
harryovo@harryovo-virtual-machine: ~/Desktop/lab1
 r11 = 0x00000000     ( decimal: 0 )
 r12 = 0x00000000     ( decimal: 0 )
 r13 = 0x00000030     ( decimal: 48          ascii: '0'     ExceptionDuringInterrupt )
 r14 = 0x00FFFE90     ( decimal: 16776848    )
 r15 = 0x00FFFE84     ( decimal: 16776836    )
=====  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   = 0x00001274     ( decimal: 4724        )
 PTBR = 0x00000000     ( decimal: 0 )
 PTLR = 0x00000000     ( decimal: 0 )
                       ---- ----  ---- ----  ---- ----  --IS PZVN
 SR   = 0x00000010  =  0000 0000  0000 0000  0000 0000  0001 0000
          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      = 30039
 Time of next serial out event     = 2147483647
   Current Time                    = 676
```

```
   Time of next serial out event     = 2147483647
     Current Time                    = 676
     Time of next event              = 5005
     Time Spent Sleeping             = 0
       Instructions Executed         = 676
   Number of Disk Reads              = 0
   Number of Disk Writes             = 0
 ==============================
 The next instruction to execute will be:
 001274: 8710000A       or      r0,0x000A,r1     ! decimal: 10, ascii: ".."
 About to execute FUNCTION ENTRY                    in nl (System.c, line 48)  time = 676
```

3. st (information of stack)

output:

```
> st
   Function/Method            Frame Addr   Execution at...
   ========================   ==========   ====================
   nl                          00FFFE90    System.c, line 48
   bar                         00FFFEAC    HelloWorld.c, line 19
   bar                         00FFFEC8    HelloWorld.c, line 21
   foo                         00FFFEE0    HelloWorld.c, line 11
   main                        00FFFEF8    HelloWorld.c, line 7
Bottom of activation frame stack!
```

4. fr (information of frame)

output:

```
> fr
=====  Frame number 0 (where StackTop = 0)  =====
Function Name:     nl
Filename:          System.c
Execution now at: line 48
Frame Addr:        00FFFE90
frameSize:         4
totalParmSize:     0
                           ==========
   sp--> -12   00FFFE84:  00000000
R.D.ptr:  -8   00FFFE88:  000012A0
    r13:  -4   00FFFE8C:  00000013
     fp:   0   00FFFE90:  00FFFEAC
RetAddr:   4   00FFFE94:  00002898
                           ==========

PARAMETERS AND LOCAL VARIABLES WITHIN THIS FRAME:
=================================================
=================================================
```

5. step (walk through the execution of an assembly language program, line-by-line.)

output:

```
Enter a command at the prompt.  Type 'quit' to exit or 'help' for
info about commands.
> s
Done!  The next instruction to execute will be:
0028A8: 8F1F0000       store   r1,[r15+0x0000]  ! decimal: 0  (PowerOnReset)
> s
Done!  The next instruction to execute will be:
0028AC: 87D00015       or      r0,0x0015,r13    ! decimal: 21, ascii: ".."
> s
Done!  The next instruction to execute will be:
0028B0: 87A04341       or      r0,0x4341,r10    ! decimal: 17217, ascii: "CA"
> s
Done!  The next instruction to execute will be:
0028B4: A0FFFF58       call    0xFFFF58         ! targetAddr = _function_3_bar
> s
Done!  The next instruction to execute will be:
                  _function_3_bar:
00280C: 54EF0000       push    r14,[--r15]
> s
Done!  The next instruction to execute will be:
002810: 67EF0000       or      r15,r0,r14
> s
Done!  The next instruction to execute will be:
002814: 54DF0000       push    r13,[--r15]
> s
Done!  The next instruction to execute will be:
002818: C0100000       sethi   0x0000,r1        ! 0x000028D0 = 10448 (_RoutineDescriptor__function_3_bar)
> s
Done!  The next instruction to execute will be:
00281C: C11028D0       setlo   0x28D0,r1
> s
Done!  The next instruction to execute will be:
002820: 541F0000       push    r1,[--r15]
> s
Done!  The next instruction to execute will be:
002824: 87100003       or      r0,0x0003,r1     ! decimal: 3, ascii: ".."
> s
Done!  The next instruction to execute will be:
                  _Label_22_1:
002828: 540F0000       push    r0,[--r15]
> s
Done!  The next instruction to execute will be:
00282C: 81110001       sub     r1,0x0001,r1     ! decimal: 1, ascii: ".."
>
```