COEN 311 Lab 1

by

Mamadou Diao Kaba

27070179

Performed on

February 11th 2021

Objectives

The objectives of this lab are first to review the materials covered on lab 0, which were the method of connecting to an ENCS Linux server and learning some basic Linux commands and text editing with nano. Secondly, it was to familiarise using the nasm assembler and the ld loader to compute a program from Intel x86 assembly language source code and to single-step through the program using the gdb command line debugger.

Theory

The Linux nasm assembler role is to assemble the instructions of the Intel x86 into the machine. The gdb debugger role is to allow for the register and memory contents to be read by single-stepping through the program. Finally, the role of the ld loader program is to convert the machine code into an executable program and to load it into the memory.

Question

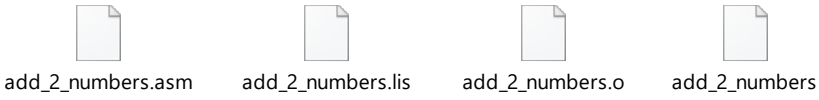
Explain the differences between an assembly language source code file, the listing file produced by nasm, the object file (.o) and the final executable program.

The assembly language source code file is the instructions given to the machines in ASCII that are understandable for humans. The listing file produced by nasm is an ASCII text file produced by nasm containing the machine code of the program on the left and the source code on the right. The object file (.o) is an output file that contains parts of the machine code that isn't fully linked into the complete program. The final executable program is a binary code that is executed by the CPU.

Conclusion

We familiarised first with the Linux nasm assembler by showing it assemble the instructions given into the machine (assembly language source code file). Secondly, with the ld loader program role to convert the machine code into an executable program and to load it into the memory (listing file produced by nasm, object file). And finally, we familiarise with the gdb debugger role to allow for the register and memory contents to be read by single-stepping through the program (final executable program).

APPENDIX



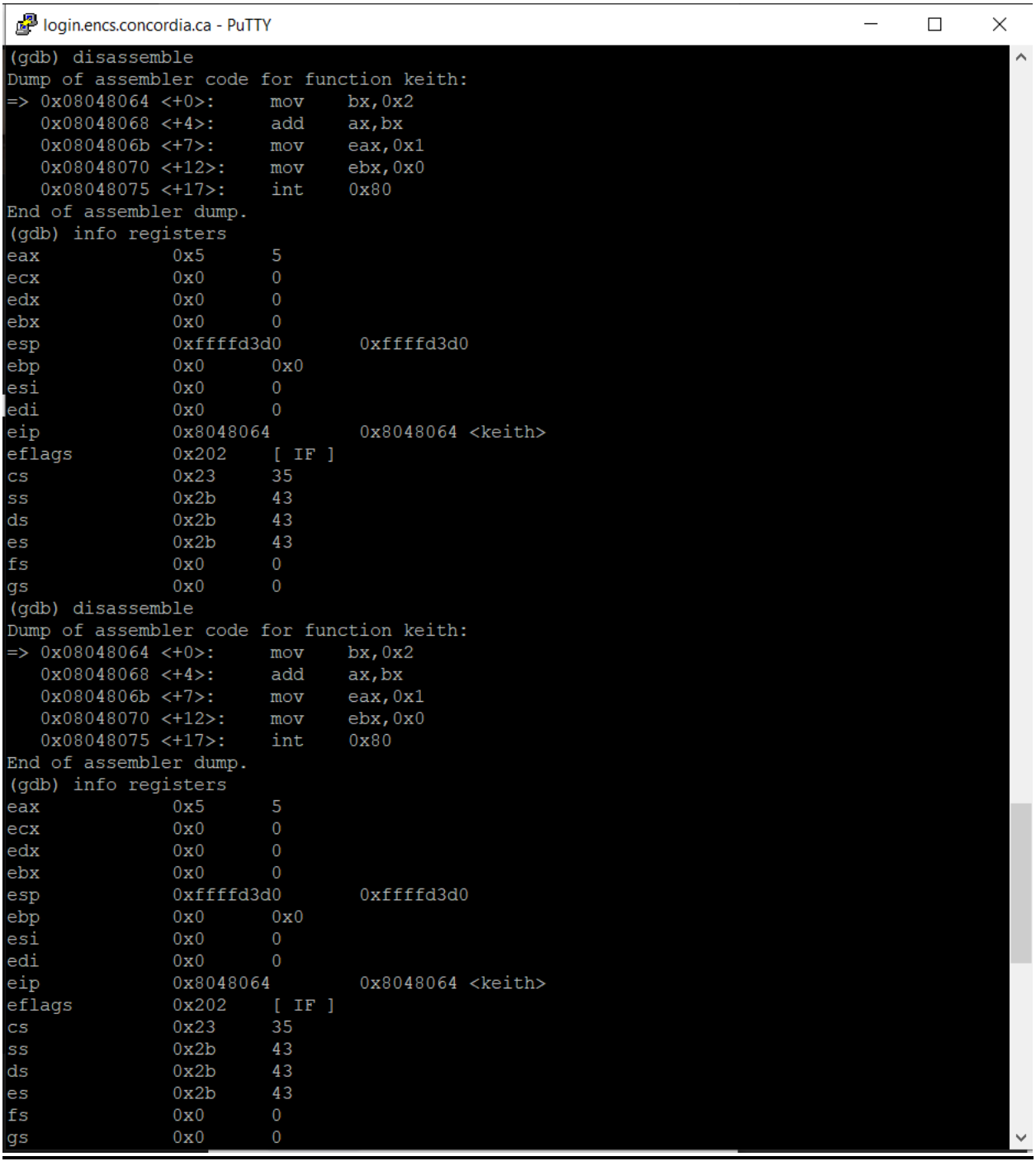
-

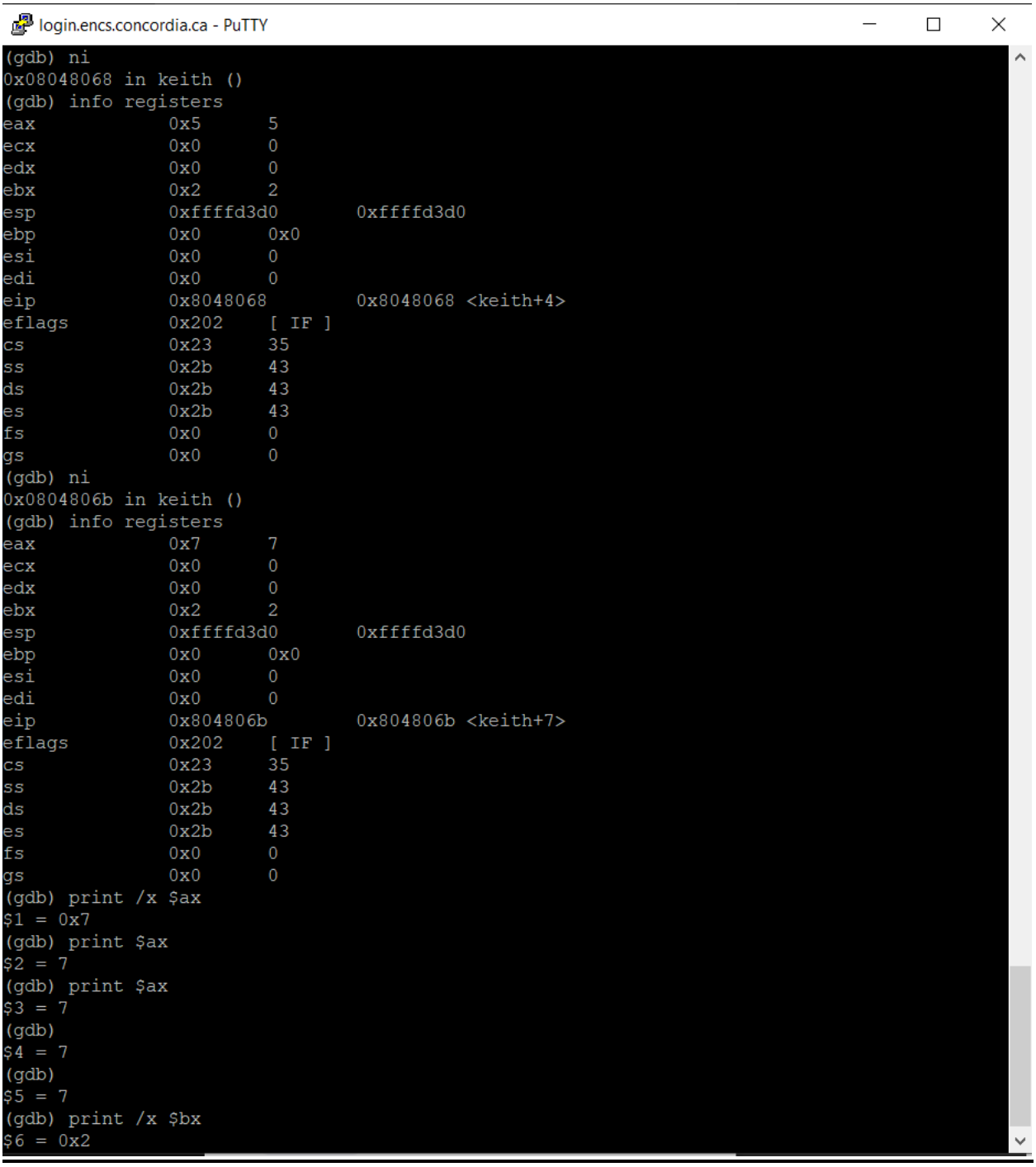
login.encs.concordia.ca - PuTTY

```
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from add_2_numbers...(no debugging symbols found)...done.
(gdb) break keith
Breakpoint 1 at 0x8048064
(gdb) run
Starting program: /nfs/home/m/ma_kaba/COEN311/NASM/lab1/add_2_numbers

Breakpoint 1, 0x08048064 in keith ()
(gdb) info registers
eax            0x5      5
ecx            0x0      0
edx            0x0      0
ebx            0x0      0
esp            0xffffd3d0       0xffffd3d0
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048064        0x8048064 <keith>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
(gdb) ni
0x08048068 in keith ()
(gdb) info registers
eax            0x5      5
ecx            0x0      0
edx            0x0      0
ebx            0x2      2
esp            0xffffd3d0       0xffffd3d0
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048068        0x8048068 <keith+4>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
```

login.encs.concordia.ca - PuTTY

```
(gdb) disassemble
Dump of assembler code for function keith:
   0x08048064 <+0>:     mov    $0x2,%bx
=> 0x08048068 <+4>:     add    %bx,%ax
   0x0804806b <+7>:     mov    $0x1,%eax
   0x08048070 <+12>:    mov    $0x0,%ebx
   0x08048075 <+17>:    int    $0x80
End of assembler dump.
(gdb) quit
A debugging session is active.

        Inferior 1 [process 37117] will be killed.

Quit anyway? (y or n) y
[poise] [/home/m/ma_kaba/COEN311/NASM/lab1] > gdb
GNU gdb (GDB) 7.7
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) quit
[poise] [/home/m/ma_kaba/COEN311/NASM/lab1] > gdb add_2_numbers
GNU gdb (GDB) 7.7
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from add_2_numbers...(no debugging symbols found)...done.
(gdb) set disassembly-flavor intel
(gdb) break keith
Breakpoint 1 at 0x8048064
(gdb) run
```


login.encs.concordia.ca - PuTTY
(gdb) disassemble
Dump of assembler code for function keith:
=> 0x08048064 <+0>:   mov   bx,0x2
   0x08048068 <+4>:   add   ax,bx
   0x0804806b <+7>:   mov   eax,0x1
   0x08048070 <+12>:   mov   ebx,0x0
   0x08048075 <+17>:   int   0x80
End of assembler dump.
(gdb) info registers
eax   0x5   5
ecx   0x0   0
edx   0x0   0
ebx   0x0   0
esp   0xffffd3d0   0xffffd3d0
ebp   0x0   0x0
esi   0x0   0
edi   0x0   0
eip   0x8048064   0x8048064 <keith>
eflags   0x202   [IF]
cs   0x23   35
ss   0x2b   43
ds   0x2b   43
es   0x2b   43
fs   0x0   0
gs   0x0   0
(gdb) disassemble
Dump of assembler code for function keith:
=> 0x08048064 <+0>:   mov   bx,0x2
   0x08048068 <+4>:   add   ax,bx
   0x0804806b <+7>:   mov   eax,0x1
   0x08048070 <+12>:   mov   ebx,0x0
   0x08048075 <+17>:   int   0x80
End of assembler dump.
(gdb) info registers
eax   0x5   5
ecx   0x0   0
edx   0x0   0
ebx   0x0   0
esp   0xffffd3d0   0xffffd3d0
ebp   0x0   0x0
esi   0x0   0
edi   0x0   0
eip   0x8048064   0x8048064 <keith>
eflags   0x202   [IF]
cs   0x23   35
ss   0x2b   43
ds   0x2b   43
es   0x2b   43
fs   0x0   0
gs   0x0   0


login.encs.concordia.ca - PuTTY
(gdb) ni
0x08048068 in keith ()
(gdb) info registers
eax   0x5   5
ecx   0x0   0
edx   0x0   0
ebx   0x2   2
esp   0xffffd3d0   0xffffd3d0
ebp   0x0   0x0
esi   0x0   0
edi   0x0   0
eip   0x8048068   0x8048068 <keith+4>
eflags   0x202   [IF]
cs   0x23   35
ss   0x2b   43
ds   0x2b   43
es   0x2b   43
fs   0x0   0
gs   0x0   0
(gdb) ni
0x0804806b in keith ()
(gdb) info registers
eax   0x7   7
ecx   0x0   0
edx   0x0   0
ebx   0x2   2
esp   0xffffd3d0   0xffffd3d0
ebp   0x0   0x0
esi   0x0   0
edi   0x0   0
eip   0x804806b   0x804806b <keith+7>
eflags   0x202   [IF]
cs   0x23   35
ss   0x2b   43
ds   0x2b   43
es   0x2b   43
fs   0x0   0
gs   0x0   0
(gdb) print /x $ax
$1 = 0x7
(gdb) print $ax
$2 = 7
(gdb) print $ax
$3 = 7
(gdb)
$4 = 7
(gdb)
$5 = 7
(gdb) print /x $bx
$6 = 0x2