전자정보공학부 20160433 김민정

1. 설치소감

-어려웠던 점 : bochs를 설치했는데 에뮬레이터 실행이 되지 않았다.

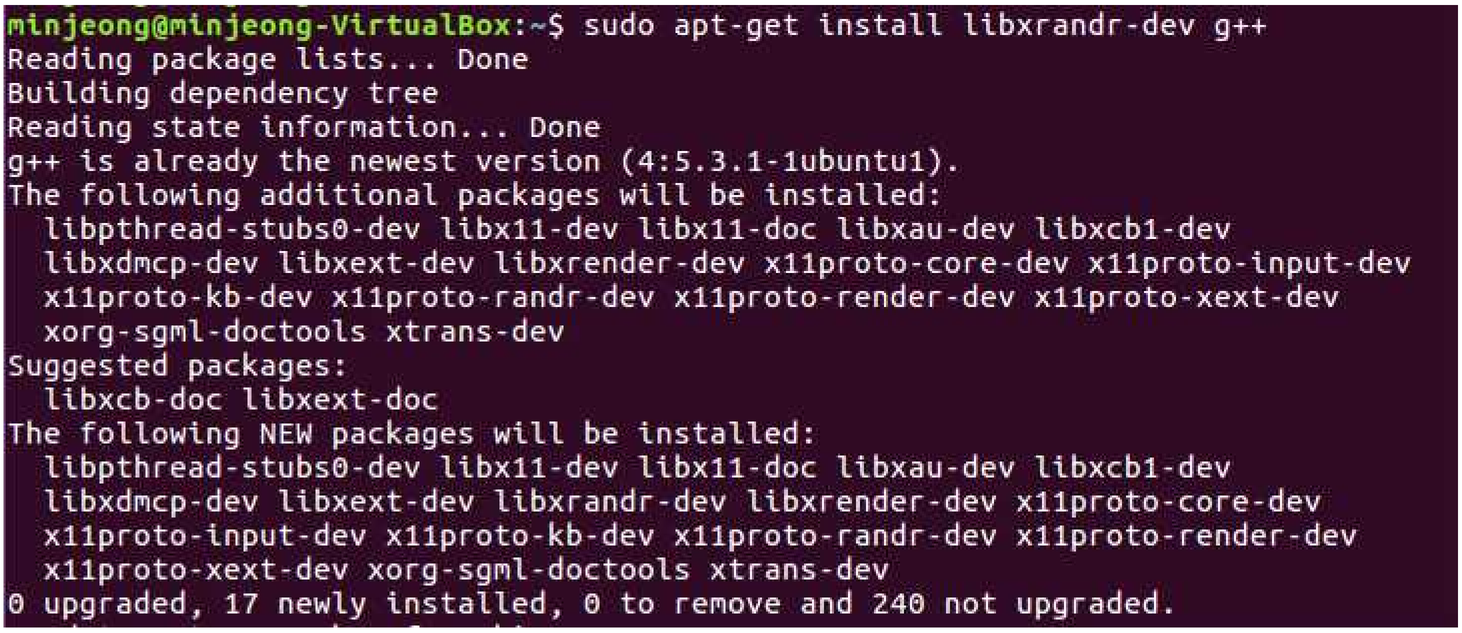
->해결 : 처음에 configure를 실행하여 설정을 추가할 때 ―with-x11을 ―with-xll로 잘못쓰는 바람에 생긴 오류였다. 이 문제를 해결하기위해 bochs설치방법을 따로 검색하여 링크파일을 생성했었는데 다시 명세를 읽으면서 x11을 xll로 썼다는 것을 깨닫고 다시 configure를 실행하여 에뮬레이터를 실행시킬 수 있었다.

-어려웠던 점 : 어셈블리코드를 수정할 때 문자를 하나씩 입력하고 싶지 않았다.

->해결 : 문자를 하나하나 입력해보던 중 규칙성이 보여 반복문을 만들기 위해 노력했다. 레지스터에 값을 집어넣는 방식이 낯설어 처음에는 헤맸지만 검색을 하여 원하는대로 값을 집어넣을 수 있었다. 그래서 MSGLOOP레이블을 반복하는 방식으로 어셈블리코드를 수정할 수 있었다.

2. 실행결과

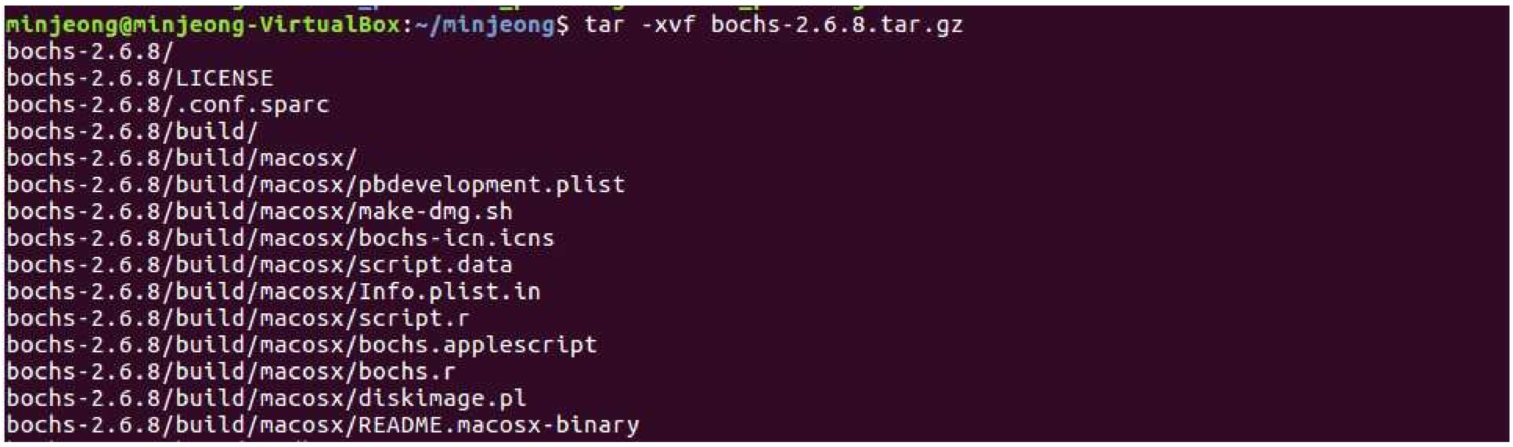
1) libxrandr-dev 및 g++ 설치



```
minjeong@minjeong-VirtualBox:~$ sudo apt-get install libxrandr-dev g++
Reading package lists... Done
Building dependency tree
Reading state information... Done
g++ is already the newest version (4:5.3.1-1ubuntu1).
The following additional packages will be installed:
  libpthread-stubs0-dev libx11-dev libx11-doc libxau-dev libxcb1-dev
  libxdmcp-dev libxext-dev libxrender-dev x11proto-core-dev x11proto-input-dev
  x11proto-kb-dev x11proto-randr-dev x11proto-render-dev x11proto-xext-dev
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  libxcb-doc libxext-doc
The following NEW packages will be installed:
  libpthread-stubs0-dev libx11-dev libx11-doc libxau-dev libxcb1-dev
  libxdmcp-dev libxext-dev libxrandr-dev libxrender-dev x11proto-core-dev
  x11proto-input-dev x11proto-kb-dev x11proto-randr-dev x11proto-render-dev
  x11proto-xext-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 17 newly installed, 0 to remove and 240 not upgraded.
```

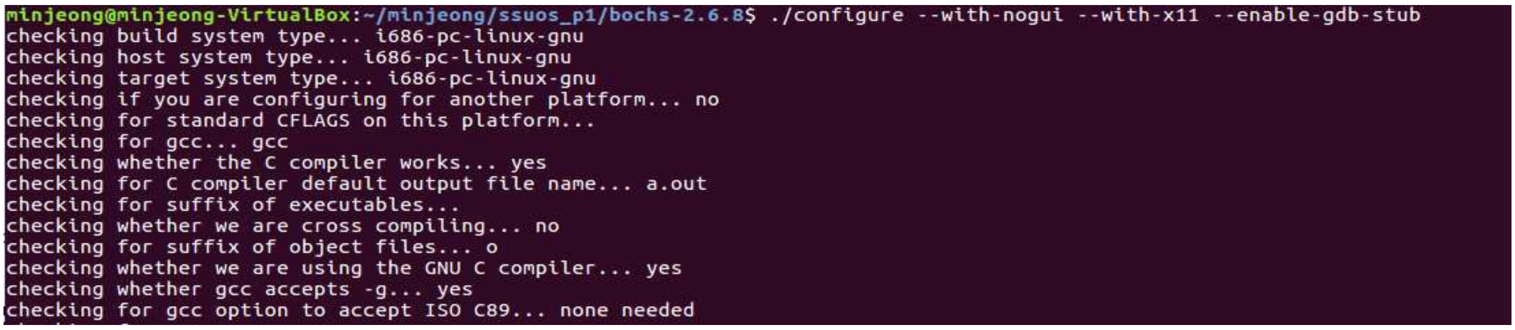
2)bochs 설치

-압축해제



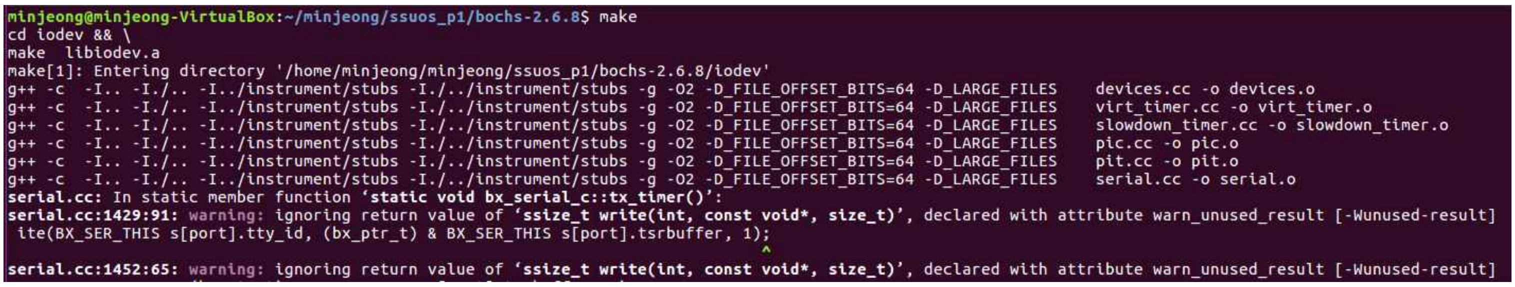
```
minjeong@minjeong-VirtualBox:~/minjeong$ tar -xvf bochs-2.6.8.tar.gz
bochs-2.6.8/
bochs-2.6.8/LICENSE
bochs-2.6.8/.conf.sparc
bochs-2.6.8/build/
bochs-2.6.8/build/macosx/
bochs-2.6.8/build/macosx/pbdevelopment.plist
bochs-2.6.8/build/macosx/make-dmg.sh
bochs-2.6.8/build/macosx/bochs-icn.icns
bochs-2.6.8/build/macosx/script.data
bochs-2.6.8/build/macosx/Info.plist.in
bochs-2.6.8/build/macosx/script.r
bochs-2.6.8/build/macosx/bochs.applescript
bochs-2.6.8/build/macosx/bochs.r
bochs-2.6.8/build/macosx/diskimage.pl
bochs-2.6.8/build/macosx/README.macosx-binary
```

-configure 설정



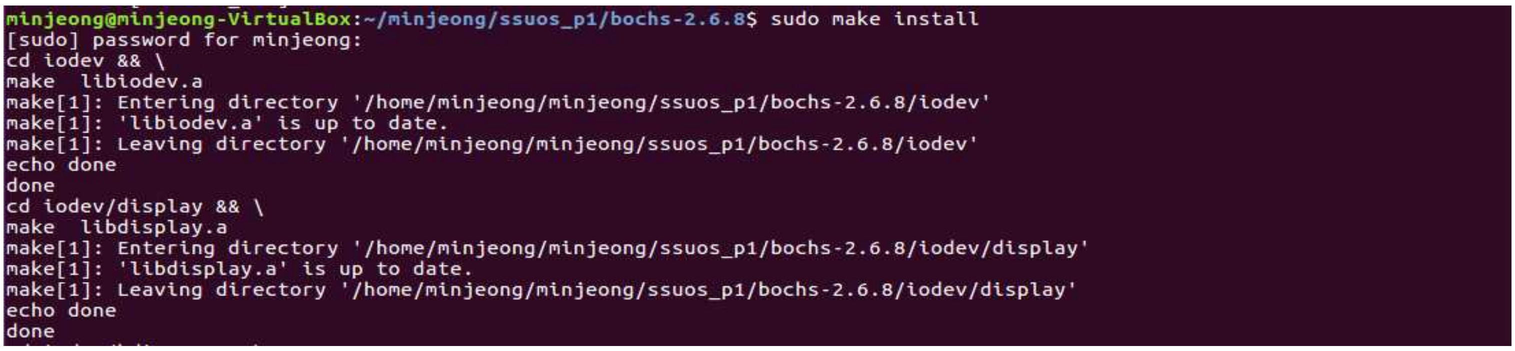
```
minjeong@minjeong-VirtualBox:~/minjeong/ssuos_p1/bochs-2.6.8$ ./configure --with-nogui --with-x11 --enable-gdb-stub
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking if you are configuring for another platform... no
checking for standard CFLAGS on this platform...
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
```

-make



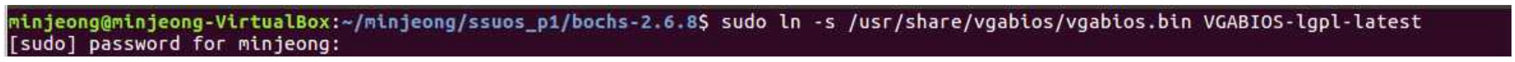
```
minjeong@minjeong-VirtualBox:~/minjeong/ssuos_p1/bochs-2.6.8$ make
cd iodev && \
make  libiodev.a
make[1]: Entering directory '/home/minjeong/minjeong/ssuos_p1/bochs-2.6.8/iodev'
g++ -c  -I.. -I./.. -I../instrument/stubs -I./../instrument/stubs -g -O2 -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES    devices.cc -o devices.o
g++ -c  -I.. -I./.. -I../instrument/stubs -I./../instrument/stubs -g -O2 -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES    virt_timer.cc -o virt_timer.o
g++ -c  -I.. -I./.. -I../instrument/stubs -I./../instrument/stubs -g -O2 -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES    slowdown_timer.cc -o slowdown_timer.o
g++ -c  -I.. -I./.. -I../instrument/stubs -I./../instrument/stubs -g -O2 -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES    pic.cc -o pic.o
g++ -c  -I.. -I./.. -I../instrument/stubs -I./../instrument/stubs -g -O2 -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES    pit.cc -o pit.o
g++ -c  -I.. -I./.. -I../instrument/stubs -I./../instrument/stubs -g -O2 -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES    serial.cc -o serial.o
serial.cc: In static member function ‘static void bx_serial_c::tx_timer()’:
serial.cc:1429:91: warning: ignoring return value of ‘ssize_t write(int, const void*, size_t)’, declared with attribute warn_unused_result [-Wunused-result]
 ite(BX_SER_THIS s[port].tty_id, (bx_ptr_t) & BX_SER_THIS s[port].tsrbuffer, 1);
                                                                              ^

serial.cc:1452:65: warning: ignoring return value of ‘ssize_t write(int, const void*, size_t)’, declared with attribute warn_unused_result [-Wunused-result]
```

-make install



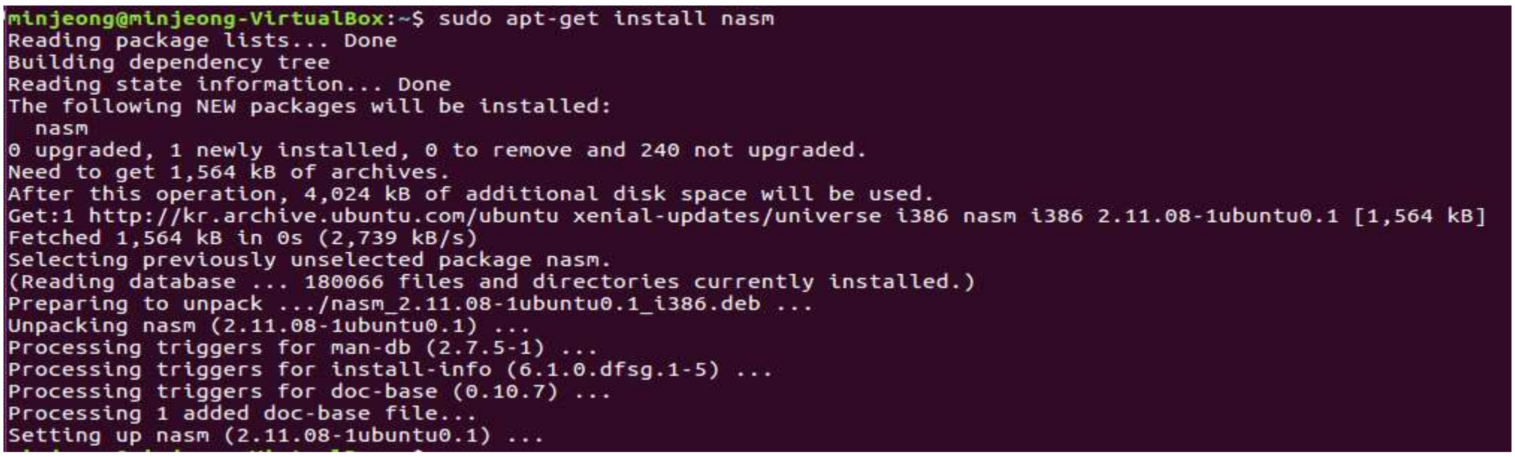
```
minjeong@minjeong-VirtualBox:~/minjeong/ssuos_p1/bochs-2.6.8$ sudo make install
[sudo] password for minjeong:
cd iodev && \
make  libiodev.a
make[1]: Entering directory '/home/minjeong/minjeong/ssuos_p1/bochs-2.6.8/iodev'
make[1]: 'libiodev.a' is up to date.
make[1]: Leaving directory '/home/minjeong/minjeong/ssuos_p1/bochs-2.6.8/iodev'
echo done
done
cd iodev/display && \
make  libdisplay.a
make[1]: Entering directory '/home/minjeong/minjeong/ssuos_p1/bochs-2.6.8/iodev/display'
make[1]: 'libdisplay.a' is up to date.
make[1]: Leaving directory '/home/minjeong/minjeong/ssuos_p1/bochs-2.6.8/iodev/display'
echo done
done
```

-링크파일생성



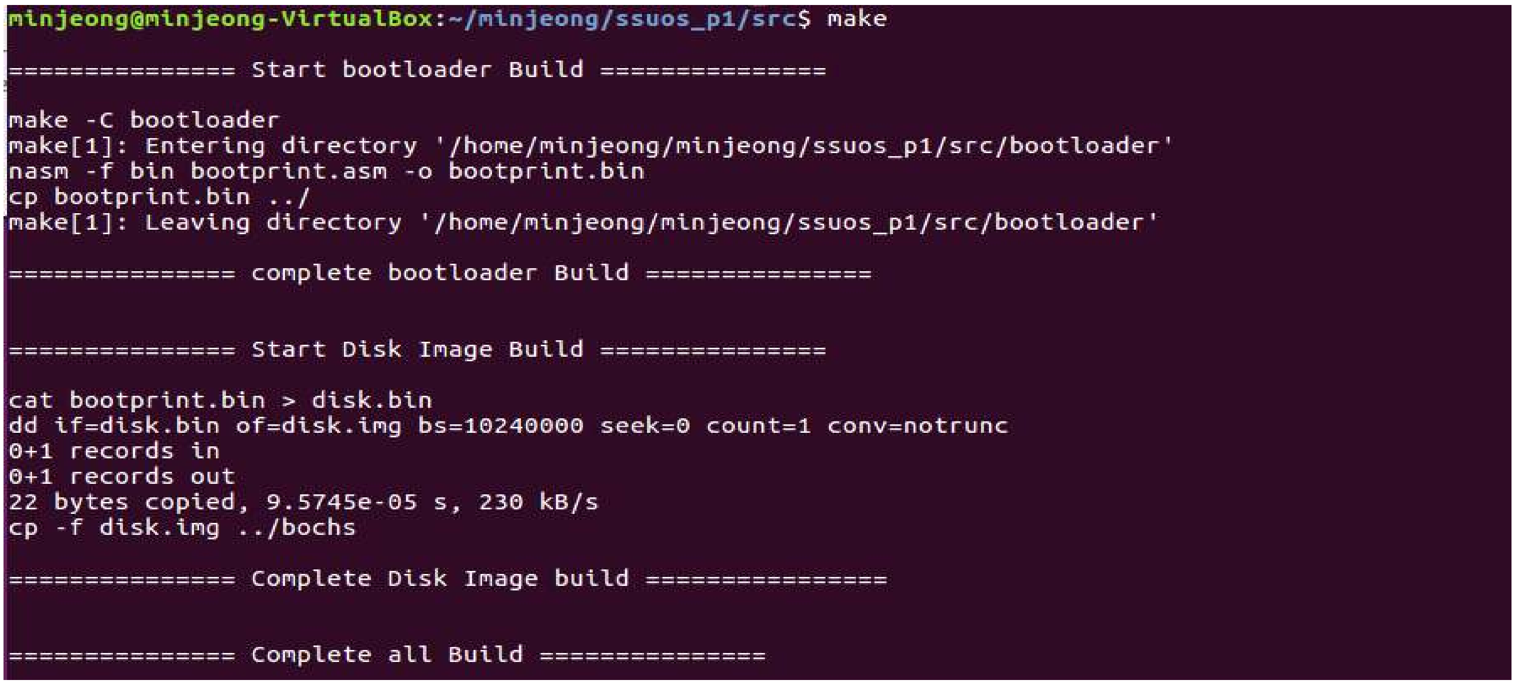
```
minjeong@minjeong-VirtualBox:~/minjeong/ssuos_p1/bochs-2.6.8$ sudo ln -s /usr/share/vgabios/vgabios.bin VGABIOS-lgpl-latest
[sudo] password for minjeong:
```

3)nasm 설치



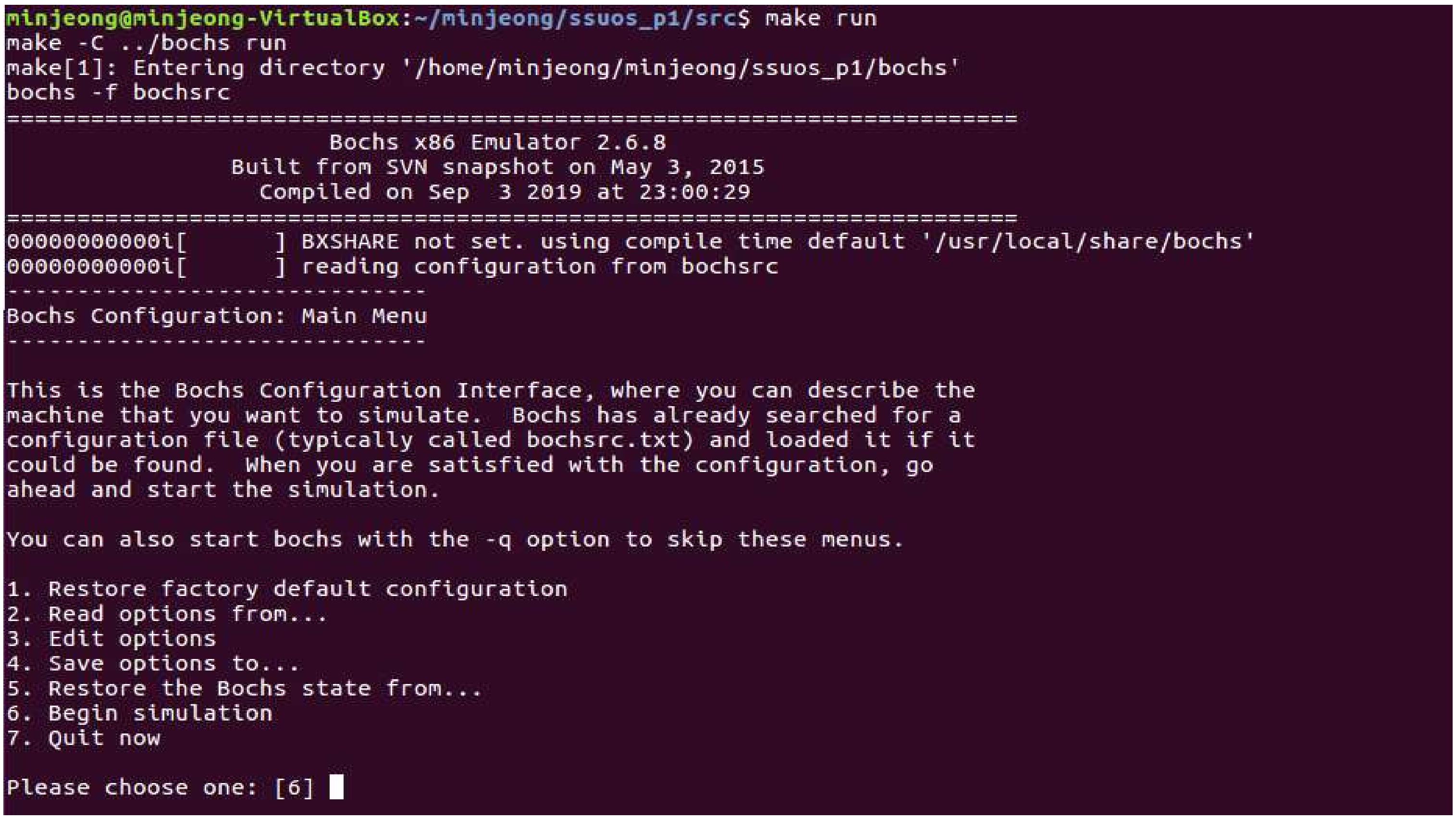
```
minjeong@minjeong-VirtualBox:~$ sudo apt-get install nasm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nasm
0 upgraded, 1 newly installed, 0 to remove and 240 not upgraded.
Need to get 1,564 kB of archives.
After this operation, 4,024 kB of additional disk space will be used.
Get:1 http://kr.archive.ubuntu.com/ubuntu xenial-updates/universe i386 nasm i386 2.11.08-1ubuntu0.1 [1,564 kB]
Fetched 1,564 kB in 0s (2,739 kB/s)
Selecting previously unselected package nasm.
(Reading database ... 180066 files and directories currently installed.)
Preparing to unpack .../nasm_2.11.08-1ubuntu0.1_i386.deb ...
Unpacking nasm (2.11.08-1ubuntu0.1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for install-info (6.1.0.dfsg.1-5) ...
Processing triggers for doc-base (0.10.7) ...
Processing 1 added doc-base file...
Setting up nasm (2.11.08-1ubuntu0.1) ...
```

4)make



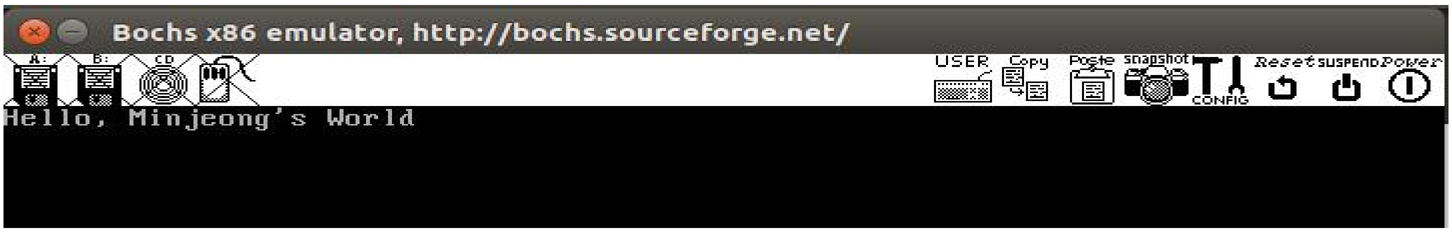
```
minjeong@minjeong-VirtualBox:~/minjeong/ssuos_p1/src$ make

=============== Start bootloader Build ===============

make -C bootloader
make[1]: Entering directory '/home/minjeong/minjeong/ssuos_p1/src/bootloader'
nasm -f bin bootprint.asm -o bootprint.bin
cp bootprint.bin ../
make[1]: Leaving directory '/home/minjeong/minjeong/ssuos_p1/src/bootloader'

=============== complete bootloader Build ===============


=============== Start Disk Image Build ===============

cat bootprint.bin > disk.bin
dd if=disk.bin of=disk.img bs=10240000 seek=0 count=1 conv=notrunc
0+1 records in
0+1 records out
22 bytes copied, 9.5745e-05 s, 230 kB/s
cp -f disk.img ../bochs

=============== Complete Disk Image build ================


=============== Complete all Build ===============
```

5)make run



```
minjeong@minjeong-VirtualBox:~/minjeong/ssuos_p1/src$ make run
make -C ../bochs run
make[1]: Entering directory '/home/minjeong/minjeong/ssuos_p1/bochs'
bochs -f bochsrc
========================================================================
                       Bochs x86 Emulator 2.6.8
                Built from SVN snapshot on May 3, 2015
                  Compiled on Sep  3 2019 at 23:00:29
========================================================================
00000000000i[      ] BXSHARE not set. using compile time default '/usr/local/share/bochs'
00000000000i[      ] reading configuration from bochsrc
------------------------------
Bochs Configuration: Main Menu
------------------------------

This is the Bochs Configuration Interface, where you can describe the
machine that you want to simulate.  Bochs has already searched for a
configuration file (typically called bochsrc.txt) and loaded it if it
could be found.  When you are satisfied with the configuration, go
ahead and start the simulation.

You can also start bochs with the -q option to skip these menus.

1. Restore factory default configuration
2. Read options from...
3. Edit options
4. Save options to...
5. Restore the Bochs state from...
6. Begin simulation
7. Quit now

Please choose one: [6] 
```

6)“Hello, Minjeong's World”를 bochs 에뮬레이터 상에 프린트



```
Hello, Minjeong's World
```

3.수정한 어셈블리코드

```
org 0x7c00 ;메모리의 몇 번지에서 실행해야 하는지를 알려주는 선언문. 부트로더는 0x7c00으로 올라간다.
[BITS 16] ;이 프로그램이 16비트 단위로 데이터를 처리하는 프로그램임을 알린다.

START:
mov             ax, 0xb800 ;비디오 메모리의 시작주소(0xb800)를 ax에 저장하여 ax레지스터를 비디오메모리로 사용한다.
mov             es, ax ;ES레지스터에 ax값을 넣는다.
mov             ax, 0x00 ;ax레지스터를 0으로 초기화한다.
mov             bx, 0 ;bx레지스터를 0으로 초기화한다.
mov             cx, 80*25*2 ;cx레지스터에 80*25*2(가로*세로*2byte[글자])값을 저장한다.

CLS:
mov      [es:bx], ax ;비디오메모리의 주소에 0으로 초기화된 ax의 값을 집어넣어 문자를 삭제한다.
add      bx, 1 ;bx를 1씩 증가시킨다.
loop     CLS ;CLS레이블로 돌아가 루프를 도는데 루프의 횟수는 cx에 저장된 값이다.

mov si,0 ;SI레지스터를 0으로 초기화한다.
mov di,0 ;DI레지스터를 0으로 초기화한다.
mov ah,0x07 ;ax레지스터의 상위1바이트에 0x07을 대입하여 글자색을 흰색으로 지정해준다.

MSGLOOP:
mov al,byte[si+MSG] ;MSG의 주소에서 SI레지스터에 저장된 값만큼을 더한 위치의 문자를 ax레지스터의 하위1바이트에 복
사한다.
cmp al,0 ;복사된 문자와 0을 비교한다.
je END ;복사한 문자가 0이면 문자열이 종료된 것이므로 END레이블로 이동하여 문자출력을 종료한다.
mov [es:di],ax ;비디오메모리의 주소에 문자를 출력한다.
add si,1 ;SI레지스터에 1을 더하여 다음 문자열로 이동한다.
add di,2 ;DI레지스터에 2를 더하여 비디오메모리의 다음문자위치로 이동한다.
jmp MSGLOOP ;MSGLOOP레이블로 이동하여 다음문자를 출력한다.

END:
jmp $ ;현재위치에서 무한루프를 수행한다.

MSG:
db "Hello, Minjeong's World",0 ;출력할 메시지를 정의. 메시지의 마지막을 0으로 설정하여 문자열의 종료를 알 수 있다.
```

```
times 510-($-$$) db 0x00 ;현재위치($(현재주소)에서 $$(처음시작주소)를 뺀 주소)에서 510까지 0으로 채운다.
dw 0xaa55 ;dw는 word단위(2byte)이고 little endian방식이므로 511번지에는 0x55가, 512번지에는 0xaa를 채워넣는다.
```