

1. Login to openlab.ics.uci.edu

```
Using username "shuqiny2".
Server refused our key
shuqiny2@openlab.ics.uci.edu's password:
##### MOTD #####

You have logged into circinus-15.ics.uci.edu
  Operating System:  CentOS Linux release 7.8.2003 (Core)
  Architecture:      x86_64
  RAM:                91.98 GiB/94.24 GiB
  Uptime:             68 days
```

2. Add a new command to .bash_profile in login directory

```
if test -f ~/.bashrc ; then
    . ~/.bashrc
fi
source ~klefstad/.bash_profile_fix
```

3. Create a folder for homework 0

```
shuqiny2@circinus-15 20:29:19 ~
$ mkdir hw
shuqiny2@circinus-15 20:29:52 ~
$ cd hw
shuqiny2@circinus-15 20:30:01 ~/hw
$ mkdir hw0
shuqiny2@circinus-15 20:30:12 ~/hw
$ cd hw0
```

4. Create main.cpp and compile with g++

```
shuqiny2@circinus-15 20:30:14 ~/hw/hw0
$ vim main.cpp
shuqiny2@circinus-15 20:32:59 ~/hw/hw0
$ g++ main.cpp
shuqiny2@circinus-15 20:33:07 ~/hw/hw0
$ ls
a.out*  main.cpp
shuqiny2@circinus-15 20:33:18 ~/hw/hw0
$ ./a.out
Hello World!
```

5. run gdb on the program and set breakpoints

```

shuqiny2@circinus-15 20:54:01 ~/hw/hw0
$ gdb a.out
GNU gdb (GDB) 7.12
Copyright (C) 2016 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-pc-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 a.out...done.
(gdb) list
1      #include <iostream>
2      using namespace std;
3
4      int main(){
5          int i = 20;
6          cout<<"Hello World!"<<endl;
7          return 0;
8      }
9
(gdb) break 6
Breakpoint 1 at 0x4007d1: file main.cpp, line 6.

```

```

(gdb) next
Hello World!
7          return 0;
(gdb) next
8      }
(gdb) next
0x00007ffff7192555 in __libc_start_main () from /lib64/libc.so.6
(gdb) next
Single stepping until exit from function __libc_start_main,
which has no line number information.
[Inferior 1 (process 11483) exited normally]

```

6. See breakpoint details and delete it

```

(gdb) info break
Num      Type             Disp Enb Address              What
1        breakpoint       keep y   0x00000000004007e2 in main() at main.cpp:6
          breakpoint already hit 1 time
(gdb) del 1

```

7. run valgrind to check for memory leaks

```

shuginy2@circinus-15 21:01:14 ~/hw/hw0
$ valgrind a.out
==13297== Memcheck, a memory error detector
==13297== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==13297== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==13297== Command: a.out
==13297==
Hello World!
==13297==
==13297== HEAP SUMMARY:
==13297==    in use at exit: 0 bytes in 0 blocks
==13297==    total heap usage: 1 allocs, 1 frees, 72,704 bytes allocated
==13297==
==13297== All heap blocks were freed -- no leaks are possible
==13297==
==13297== For lists of detected and suppressed errors, rerun with: -s
==13297== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
shuginy2@circinus-15 21:01:38 ~/hw/hw0
$ valgrind --tool=memcheck --leak-check=yes --show-reachable=yes --track-origins
=yes a.out
==13835== Memcheck, a memory error detector
==13835== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==13835== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==13835== Command: a.out
==13835==
Hello World!
==13835==
==13835== HEAP SUMMARY:
==13835==    in use at exit: 0 bytes in 0 blocks
==13835==    total heap usage: 1 allocs, 1 frees, 72,704 bytes allocated
==13835==
==13835== All heap blocks were freed -- no leaks are possible
==13835==
==13835== For lists of detected and suppressed errors, rerun with: -s
==13835== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

8. Create a Makefile

```

XXFLAGS=-g -gdb -std=c++11 -Wpedantic -Wall -Wextra -Werror -Wzero-as-null-pointer-constant
main: main.cpp
    g++ $(CXXFLAGS) main.cpp -o main
clean:
    echo -----removing executable program main-----
    rm bin/main

```

9. Use make to build program and run it by calling main

```

shuginy2@circinus-15 21:10:28 ~/hw/hw0
$ make
g++ main.cpp -o main
shuginy2@circinus-15 21:10:32 ~/hw/hw0
$ main
Hello World!

```

10. Make clean to remove the executable "main"

```
shuqiny2@circinus-15 22:43:46 ~/hw/hw0
$ make
g++ main.cpp -o main
shuqiny2@circinus-15 22:43:51 ~/hw/hw0
$ ls
a.out* main* main.cpp Makefile .Makefile.swp
shuqiny2@circinus-15 22:43:58 ~/hw/hw0
$ make clean
echo -----removing executable program main-----
-----removing executable program main-----
/bin/rm main
shuqiny2@circinus-15 22:44:03 ~/hw/hw0
$ ls
a.out* main.cpp Makefile .Makefile.swp
```