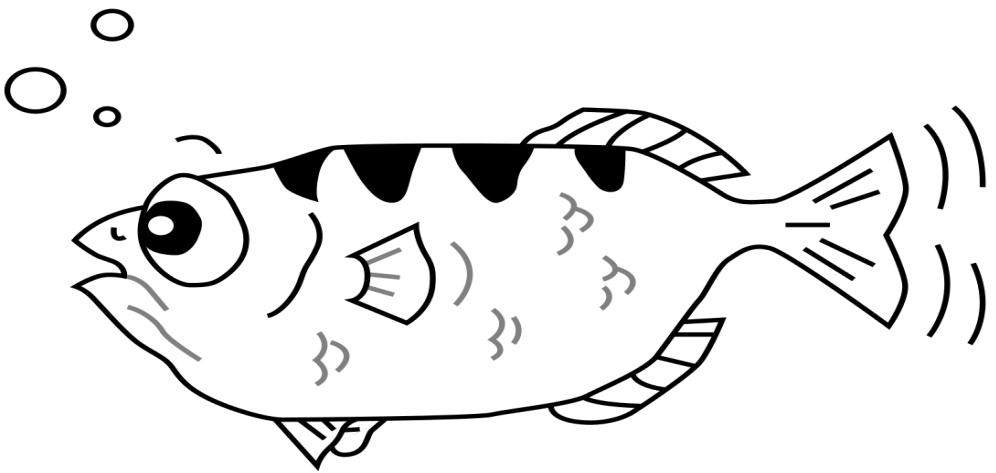
C++ & GDB & Emacs exercise

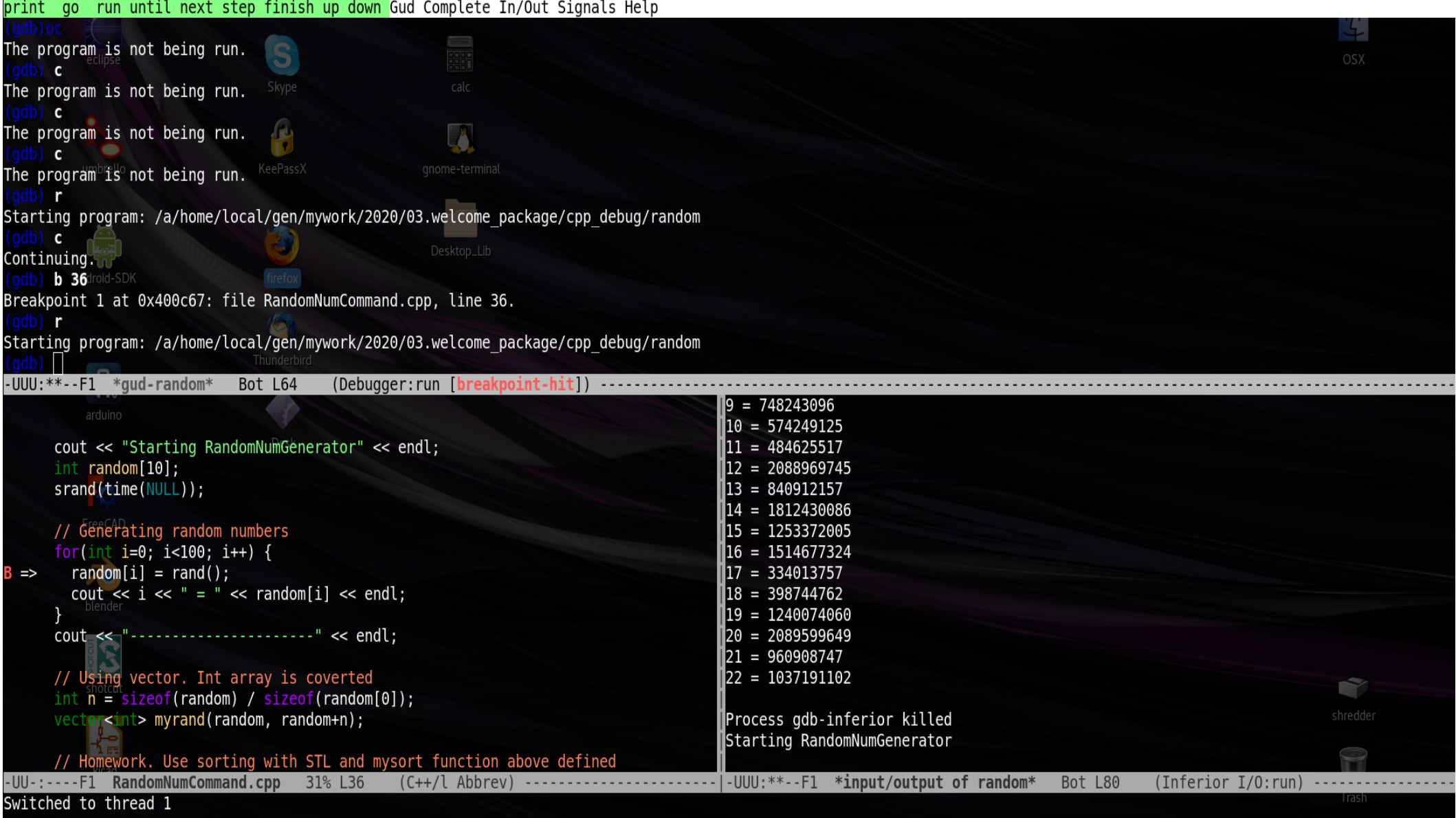
Debugger

- Cheat_GDB.txt
- GDB cheatsheet



Short exercise of GDB

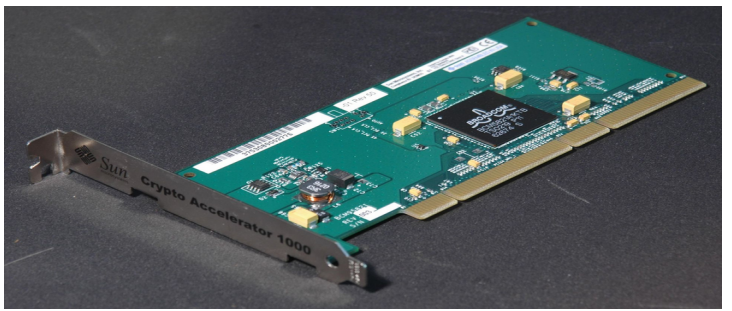
- Exercise Repository→ cpp_debug
  - Clone the code
  - Use the above codes and compile them
  - Open Emacs
  - M-x gdb
  - Set a break point
  - Run GDB commands


print go run until next step finish up down Gud Complete In/Out Signals Help
(gdb)oc
The program is not being run.
(gdb) c
The program is not being run.
(gdb) c
The program is not being run.
(gdb) c
The program is not being run.
(gdb) r
Starting program: /a/home/local/gen/mywork/2020/03.welcome_package/cpp_debug/random
(gdb) c
Continuing.
(gdb) b 36
Breakpoint 1 at 0x400c67: file RandomNumCommand.cpp, line 36.
(gdb) r
Starting program: /a/home/local/gen/mywork/2020/03.welcome_package/cpp_debug/random
(gdb)
-UUU:**--F1 *gud-random* Bot L64 (Debugger:run [breakpoint-hit])
cout << "Starting RandomNumGenerator" << endl;
int random[10];
srand(time(NULL));
// Generating random numbers
for(int i=0; i<100; i++) {
B => random[i] = rand();
cout << i << " = " << random[i] << endl;
}
cout << "----------" << endl;
// Using vector. Int array is coverted
int n = sizeof(random) / sizeof(random[0]);
vector<int> myrand(random, random+n);
// Homework. Use sorting with STL and mysort function above defined
-UU-:----F1 RandomNumCommand.cpp 31% L36 (C++/l Abbrev)
9 = 748243096
10 = 574249125
11 = 484625517
12 = 2088969745
13 = 840912157
14 = 1812430086
15 = 1253372005
16 = 1514677324
17 = 334013757
18 = 398744762
19 = 1240074060
20 = 2089599649
21 = 960908747
22 = 1037191102
Process gdb-inferior killed
Starting RandomNumGenerator
-UUU:**--F1 *input/output of random* Bot L80 (Inferior I/O:run)
Switched to thread 1

Use-case?

- Assessing hardware random number generator
  - Pseudo RN is a software

Command receiver



- Cloud Quantum Computing thread reader

Command receiver

Thread pool