Algorithm Project #1

Sorting

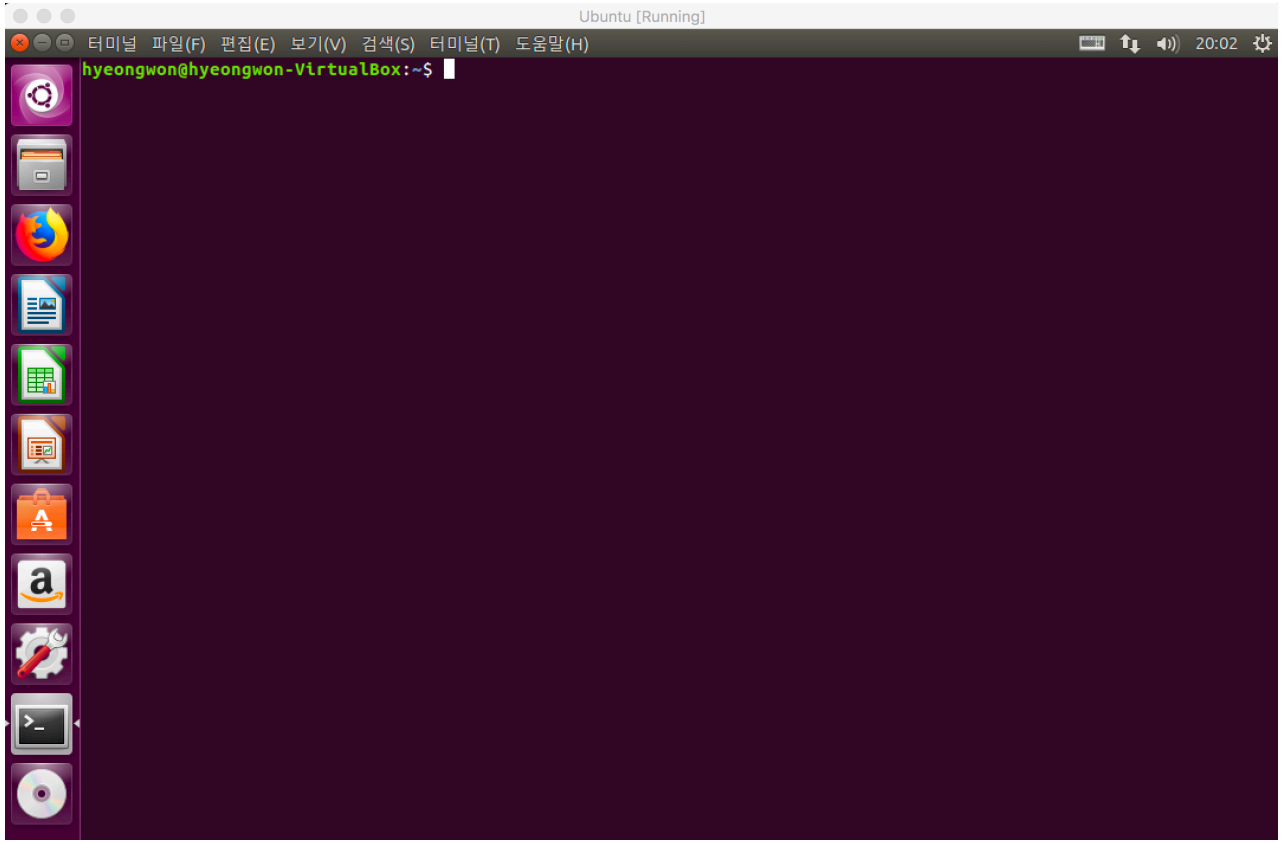
Scalable Computing Systems Laboratory
Hanyang University

Prerequisites

1. Make coding environment
   - Installing Ubuntu (Linux) on Virtual Machine (piazza)
2. Get used to using Linux commands and Vim editor.
   - Linux Commands and Vim Editor (piazza)
3. Compile source code to make executable program.
   - Just type “make”. (don’t need to know detail)
4. Submit your project
   - Just upload the zip file which contains all files in the given material. (via portal)
   - Zip file must be [Student ID]_Project1
     - Ex) 2017123456_Project1

Getting Started

1. Open the terminal: [Ctrl + Alt + t]



Download project

```
Type those lines in order from terminal.

$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install vim git
$ git clone --depth=1 https://github.com/amix/vimrc.git ~/.vim_runtime
$ sh ~/.vim_runtime/install_basic_vimrc.sh

$ wget https://piazza.com/class_profile/get_resource/je9dt75itx64ef/jfdotxcg7m96fo
$ mv jfdotxcg7m96fo sorting.zip
$ unzip sorting.zip
$ cd sorting
$ ls
```

Project Specification

```
hyeonseok@node01:~$ unzip sorting.zip
Archive:  sorting.zip
   creating: sorting/
  inflating: sorting/input-large.txt
  inflating: sorting/input-small.txt
  inflating: sorting/Makefile
  inflating: sorting/bubblesort.c
  inflating: sorting/heapsort.c
  inflating: sorting/mergesort.c
  inflating: sorting/sorting.h
  inflating: sorting/main.c
hyeonseok@node01:~$ ls sorting
Makefile      heapsort.c       input-small.txt  mergesort.c
bubblesort.c  input-large.txt  main.c           sorting.h
hyeonseok@node01:~$ cd sorting/
hyeonseok@node01:~/sorting$
```

Project Specification

- Now you can start programming!

```
$ vi main.c
$ make
$ time ./main b
```

- You can write source code main.c using Vim Editor.
- You can compile your code with just typing ‘make’.
- You can run your program and also measure the running time.

Project Specification

```
hyeonseok@node01:~$ ls sorting
Makefile      heapsort.c       input-small.txt  mergesort.c
bubblesort.c  input-large.txt  main.c           sorting.h
hyeonseok@node01:~$ cd sorting/
hyeonseok@node01:~/sorting$ vi main.c
hyeonseok@node01:~/sorting$ make
gcc -c -g -O0 main.c
gcc -c -g -O0 bubblesort.c
gcc -c -g -O0 mergesort.c
gcc -c -g -O0 heapsort.c
gcc -o main main.o bubblesort.o mergesort.o heapsort.o
hyeonseok@node01:~/sorting$ time ./main b
size = 100000
Bubble Sort Starts!

real    0m27.511s
user    0m27.504s
sys     0m0.000s
```

Project Specification

- Let’s comment out LARGE_TEST in main.c

```
$ vi main.c
```

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "sorting.h"
4 #define LARGE_TEST
5
```

- Compile and run again.

```
$ make clean
$ make
$ time ./main b
```

Project Specification

- It will cost a lot of time!
- Because sorting algorithm (function bubbleSort()) is very slow when data is large.

```
hyeonseok@node01:~/sorting$ make clean
rm -rf main main.o bubblesort.o mergesort.o heapsort.o output.txt
hyeonseok@node01:~/sorting$ make
gcc -c -g -O0 main.c
gcc -c -g -O0 bubblesort.c
gcc -c -g -O0 mergesort.c
gcc -c -g -O0 heapsort.c
gcc -o main main.o bubblesort.o mergesort.o heapsort.o
hyeonseok@node01:~/sorting$ time ./main b
size = 100000
Bubble Sort Starts!

real    0m27.504s
user    0m27.496s
sys     0m0.004s
```

Project Specification

- Your goal is to implement sorting algorithms you learned.
  - implement mergeSort, heapSort function.
    - Template code is already implemented in mergesort.c and heapsort.c
  - I strongly recommend not editing Makefile or main() function !!

```
hyeonseok@node01:~/sorting$ time ./main h
size = 100000
Heap Sort Starts!

real    0m0.095s
user    0m0.088s
sys     0m0.004s
hyeonseok@node01:~/sorting$ time ./main m
size = 100000
Merge Sort Starts!

real    0m0.090s
user    0m0.080s
sys     0m0.008s
```

output.txt

```
1 100000
2 55871
3 103930
4 105530
5 105570
6 136695
7 175440
8 185487
9 186003
10 197369
11 248203
12 258290
13 308067
14 310710
15 319153
16 331986
17 334248
18 357423
```

Submission

- We will score your code with those standards.
  - Your program must sort array correctly.
  - Your program should be completed less than 1~2 seconds.
  - Detailed description of your source code. (comments are always helpful)
- Submit your completed version of main.c file to portal.
- Note that your main.c should work with given Makefile. (You don't have to edit Makefile at all !!)

```
hyeonseok@node01:~$ zip -r 2017102943_Project.zip sorting/*
  adding: sorting/Makefile (deflated 44%)
  adding: sorting/bubblesort.c (deflated 49%)
  adding: sorting/heapsort.c (deflated 65%)
  adding: sorting/input-large.txt (deflated 52%)
  adding: sorting/input-small.txt (deflated 37%)
  adding: sorting/main.c (deflated 66%)
  adding: sorting/mergesort.c (deflated 58%)
  adding: sorting/sorting.h (deflated 51%)
hyeonseok@node01:~$ ls -l 2017102943_Project.zip
-rw-r--r-- 1 hyeonseok hyeonseok 503499 Mar 30 20:50 2017102943_Project.zip
```

Thank You