

Appendix A

Compiling Assembly Programs

A.1 Compiling on the Server

A.1.1 Connecting via SSH

You can connect to the server via the SSH (Secure SHell) service.

Using Linux or Mac OS X

If you are using Mac OS X or Linux, you can open a terminal application and type

```
$ ssh cholwich@css224.cholwich.net
```

You need to use your own account instead of “cholwich”.

Using Microsoft Windows

If you are using Microsoft Windows, you need an SSH client called “PuTTY”. It can be downloaded from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. You can run the downloaded executable (putty.exe) file with installation. When PuTTY starts, it shows the configuration window. Fill in the Host Name with `css224.cholwich.net`, and click “Open” as shown in Figure A.1.

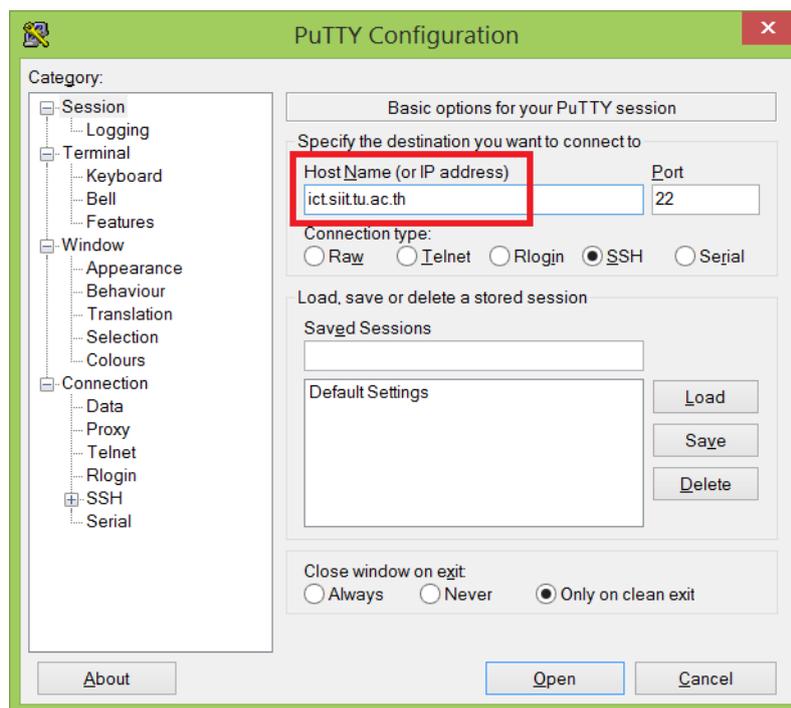


Figure A.1: PuTTY configuration window

If it is the first time that you are connecting to the ICT server, you will be asked to confirm the server’s host key. You can just click “Yes” to close this dialog as shown in Figure A.2.

Then, PuTTY will show the terminal window. Here, you will be asked for your account and password as shown in Figure A.3.



Figure A.2: PuTTY server's host key confirmation

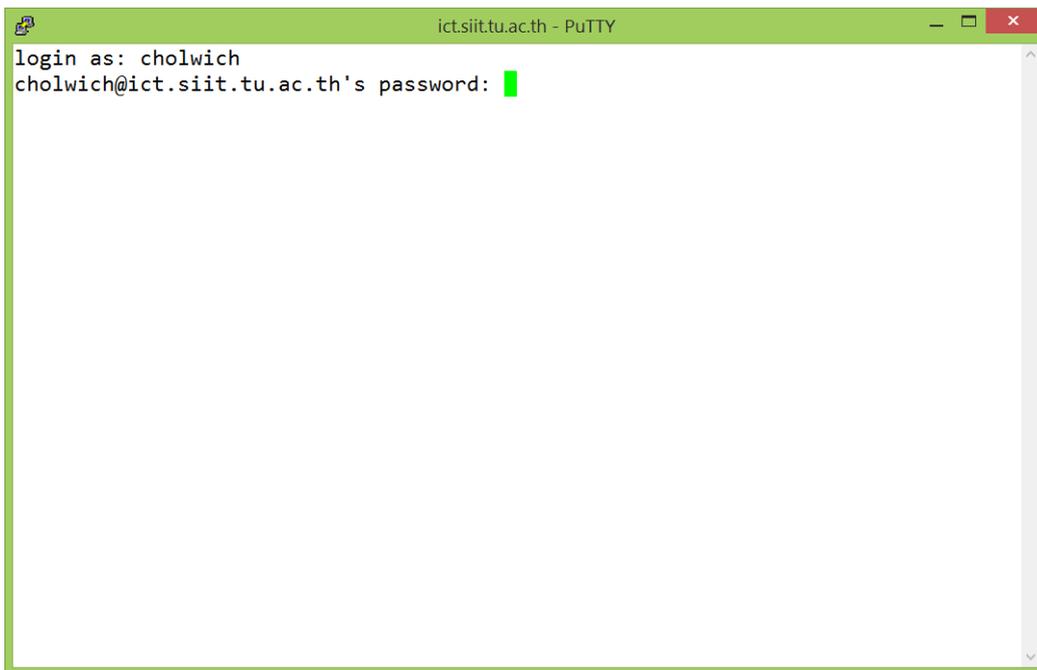


Figure A.3: PuTTY terminal window

A.1.2 Editing a source file

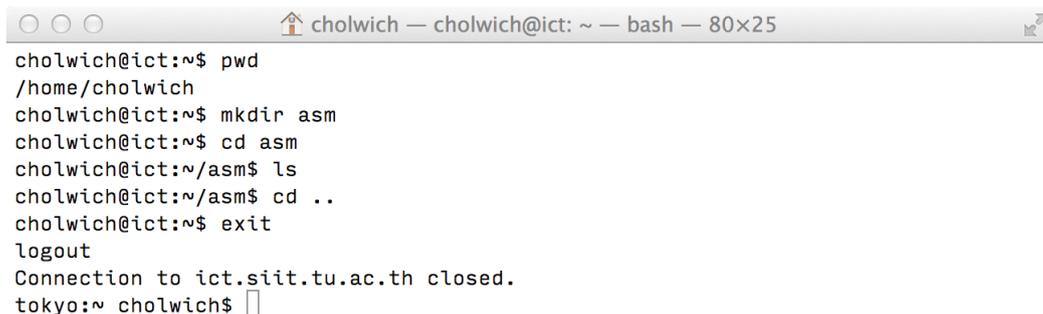
After logging into the server, your terminal application shows a command prompt waiting for you to input shell commands. Table A.1 shows a list of basic commands that you may need to manage your

source files. Figure A.4 shows a basic Linux command usage.

If you are not familiar with Linux commands, you are recommended to watch videos provided by Dr. Steven Gordon on YouTube at <http://www.youtube.com/watch?v=a8PsnTgS20A> and <http://www.youtube.com/watch?v=KVH3dMWefWE>.

Command	Description
<code>pwd</code>	Show the current directory.
<code>ls</code>	Show the list of files in the current directory.
<code>mkdir DIRECTORY</code>	Create an empty directory named DIRECTORY under the current directory.
<code>cd DIRECTORY</code>	Change the current directory to DIRECTORY . If DIRECTORY is <code>..</code> , it changes to the parent directory of the current one.
<code>cp SOURCE DEST</code>	Copy file(s) from SOURCE to DEST .
<code>mv SOURCE DEST</code>	Move file(s) from SOURCE to DEST .
<code>clear</code>	Clear the screen.
<code>exit</code>	Exit from the server.

Table A.1: Basic Linux Commands



```
cholwich@ict:~$ pwd
/home/cholwich
cholwich@ict:~$ mkdir asm
cholwich@ict:~$ cd asm
cholwich@ict:~/asm$ ls
cholwich@ict:~/asm$ cd ..
cholwich@ict:~$ exit
logout
Connection to ict.sit.tu.ac.th closed.
tokyo:~ cholwich$
```

Figure A.4: Basic Linux command usage

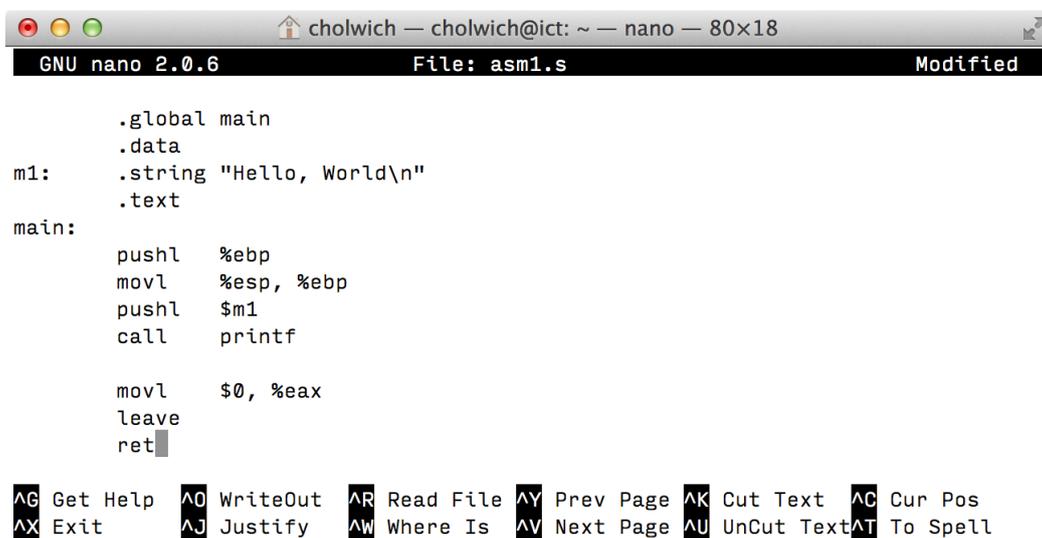
Nano¹ is a good editor for beginners. You can use it to edit your assembly source file. The editor can be started by using a command

¹<http://www.nano-editor.org>

`nano` and a file name. It is recommended that an assembly source file is ended with `.s`. For example,

```
cholwich@ict$ nano asm1.s
```

Nano will start editing a new file named `asm.s` as shown in Figure A.5. You can start typing your program. Press `Ctrl-O` to save the file, and press `Ctrl-X` to exit from Nano.



```
cholwich — cholwich@ict: ~ — nano — 80x18
GNU nano 2.0.6 File: asm1.s Modified

.global main
.data
m1: .string "Hello, World\n"
.text
main:
    pushl   %ebp
    movl   %esp, %ebp
    pushl   $m1
    call   printf

    movl   $0, %eax
    leave
    ret

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figure A.5: Nano editor usage

A.1.3 Transferring a source file to the server

Instead of editing a source file using Nano, you can create or edit the file on your local machine before transferring it to the server. You need to use `scp` command if you are using Mac OS X and Linux, and `winscp` if you are using Microsoft Windows. Figure A.6 shows how to use `scp` command. The command in the figure copies a file named `asm1.s` to the directory named `asm` under the home directory of user `cholwich` on `css224.cholwich.net`.

A terminal window screenshot showing a file transfer using scp. The window title is 'cholwich — cholwich@ict: ~ — bash — 80x10'. The terminal text shows the command 'scp asm1.s cholwich@ict.siiit.tu.ac.th:asm' being executed, with progress indicators for 'asm1.s' showing 100% completion, 144 bytes transferred, and a speed of 0.1KB/s in 00:00. The prompt returns to 'tokyo:~ cholwich\$'.Figure A.6: `scp` usage

A.1.4 Compiling an assembly file

To compile your assembly file, you need a program called `gcc` on the ICT server. Here is the basic usage:

```
gcc -o OUTPUT SOURCE1 SOURCE2
```

where `OUTPUT` is the name of the executable file, `SOURCE1` and `SOURCE2` are the names of the source files (ended with `.s` and `.c`).

To run the compiled program, use `./OUTPUT` where `OUTPUT` is the file output from `gcc`.

A.2 Compiling on an Ubuntu Linux machine

If you are using Ubuntu Linux as a default operating system or on a virtual machine, you can install `gcc` on your machine for compiling assembly programs by installing a package called `build-essential`.

```
$ sudo apt-get update  
$ sudo apt-get install build-essential
```