

Lab 1

*School of Architecture, Civil and
Environmental Engineering*

EPFL, SS 2018-2019

http://disal.epfl.ch/teaching/signals_instruments_systems/

Lab 1 outline

- This lab has two main goals:
 - Linux Revisited
 - Practicing basic C programming
- Tools:
 - gcc (C compiler)
 - Make tool

Linux commands

File permissions: `chmod <mode> <filename>`

```
chris@disallap14: ~/homework
File Edit View Terminal Help
chris@disallap14:~/homework$ ls -l
total 16
-rwxr-xr-x 1 chris chris 8471 2011-02-24 00:30 hello_world
-rw-r--r-- 1 chris chris  80 2011-02-24 00:30 hello_world.c
chris@disallap14:~/homework$
```

r: read

w: write

x: execute

Owner

Group

Other users



rwx

rwx

rwx

Linux commands

File permissions: `chmod <mode> <filename>`

Where `<mode>` is of the format:

u: file owner		r: read
g: file group		w: write
o: other users	+ -	x: execute

Examples

- Make file executable for everyone
 - `chmod ugo+x filename.txt`
- Make file unreadable by other users
 - `chmod o-r filename.txt`

Linux commands

- `ps`: displays the current programs (**p**rocesses) that are running (similar to the “task manager” of Windows)

```
user@hardy_disal:~$ ps
  PID TTY          TIME CMD
 5490 pts/0        00:00:00 bash
 5805 pts/0        00:00:15 a_program
 5808 pts/0        00:00:00 ps
```

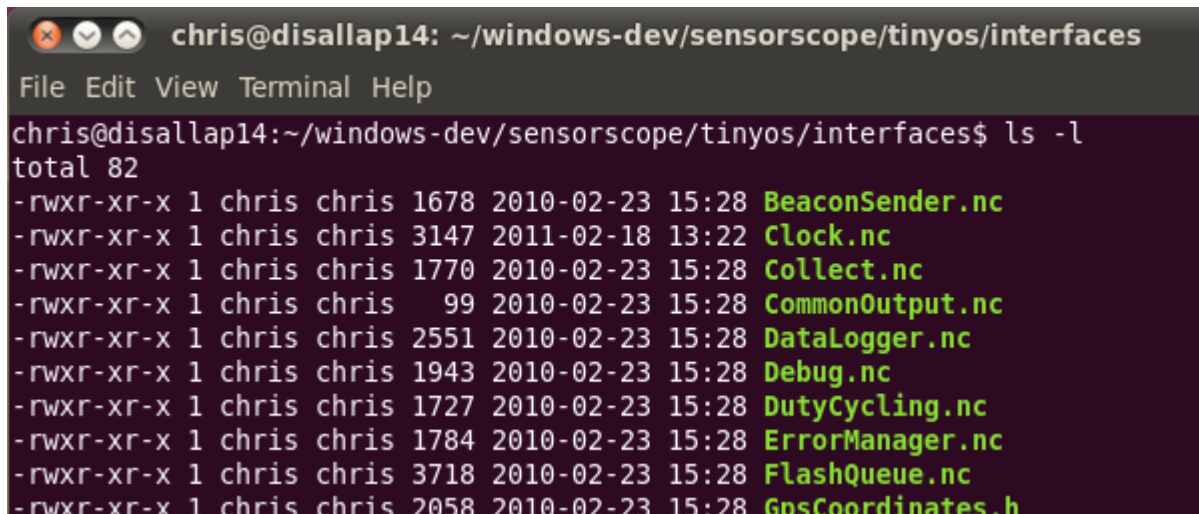
- `kill`: stop the program with the specified PID (process ID).

```
user@hardy_disal:~$ kill 5805
user@hardy_disal:~$ ps
  PID TTY          TIME CMD
 5490 pts/0        00:00:00 bash
 5809 pts/0        00:00:00 ps
```

Linux commands

Searching for text:

```
<command> | grep <search term>
```

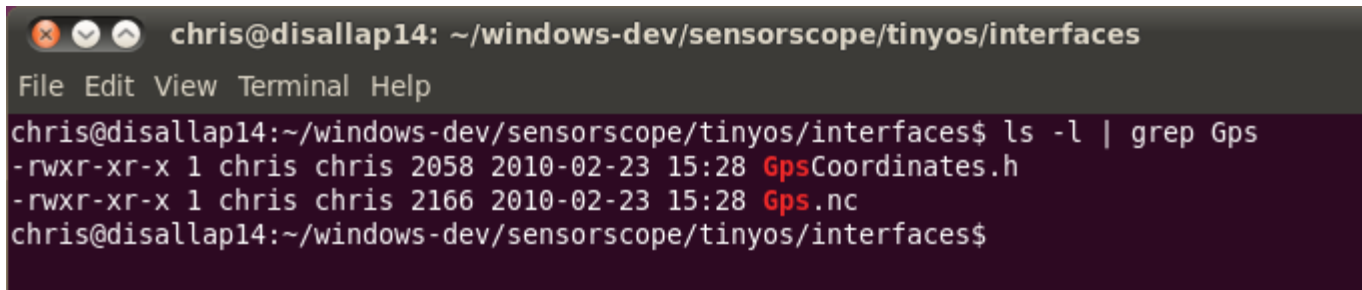


```
chris@disallap14: ~/windows-dev/sensorscope/tinyos/interfaces
File Edit View Terminal Help
chris@disallap14:~/windows-dev/sensorscope/tinyos/interfaces$ ls -l
total 82
-rwxr-xr-x 1 chris chris 1678 2010-02-23 15:28 BeaconSender.nc
-rwxr-xr-x 1 chris chris 3147 2011-02-18 13:22 Clock.nc
-rwxr-xr-x 1 chris chris 1770 2010-02-23 15:28 Collect.nc
-rwxr-xr-x 1 chris chris 99 2010-02-23 15:28 CommonOutput.nc
-rwxr-xr-x 1 chris chris 2551 2010-02-23 15:28 DataLogger.nc
-rwxr-xr-x 1 chris chris 1943 2010-02-23 15:28 Debug.nc
-rwxr-xr-x 1 chris chris 1727 2010-02-23 15:28 DutyCycling.nc
-rwxr-xr-x 1 chris chris 1784 2010-02-23 15:28 ErrorManager.nc
-rwxr-xr-x 1 chris chris 3718 2010-02-23 15:28 FlashQueue.nc
-rwxr-xr-x 1 chris chris 2058 2010-02-23 15:28 GnsCoordinates.h
```

Linux commands

Searching for text:

```
<command> | grep <search term>
```



```
chris@disallap14: ~/windows-dev/sensorscope/tinyos/interfaces
File Edit View Terminal Help
chris@disallap14:~/windows-dev/sensorscope/tinyos/interfaces$ ls -l | grep Gps
-rwxr-xr-x 1 chris chris 2058 2010-02-23 15:28 GpsCoordinates.h
-rwxr-xr-x 1 chris chris 2166 2010-02-23 15:28 Gps.nc
chris@disallap14:~/windows-dev/sensorscope/tinyos/interfaces$
```

Note: the pipe redirects output from one command to another

Compilation of C Code

- Headers contain definitions, C files contain actual implementation
- Libraries provide basic functionalities

e.g. `stdio.h` provides `printf()` function

- To compile C code, we will use `gcc`

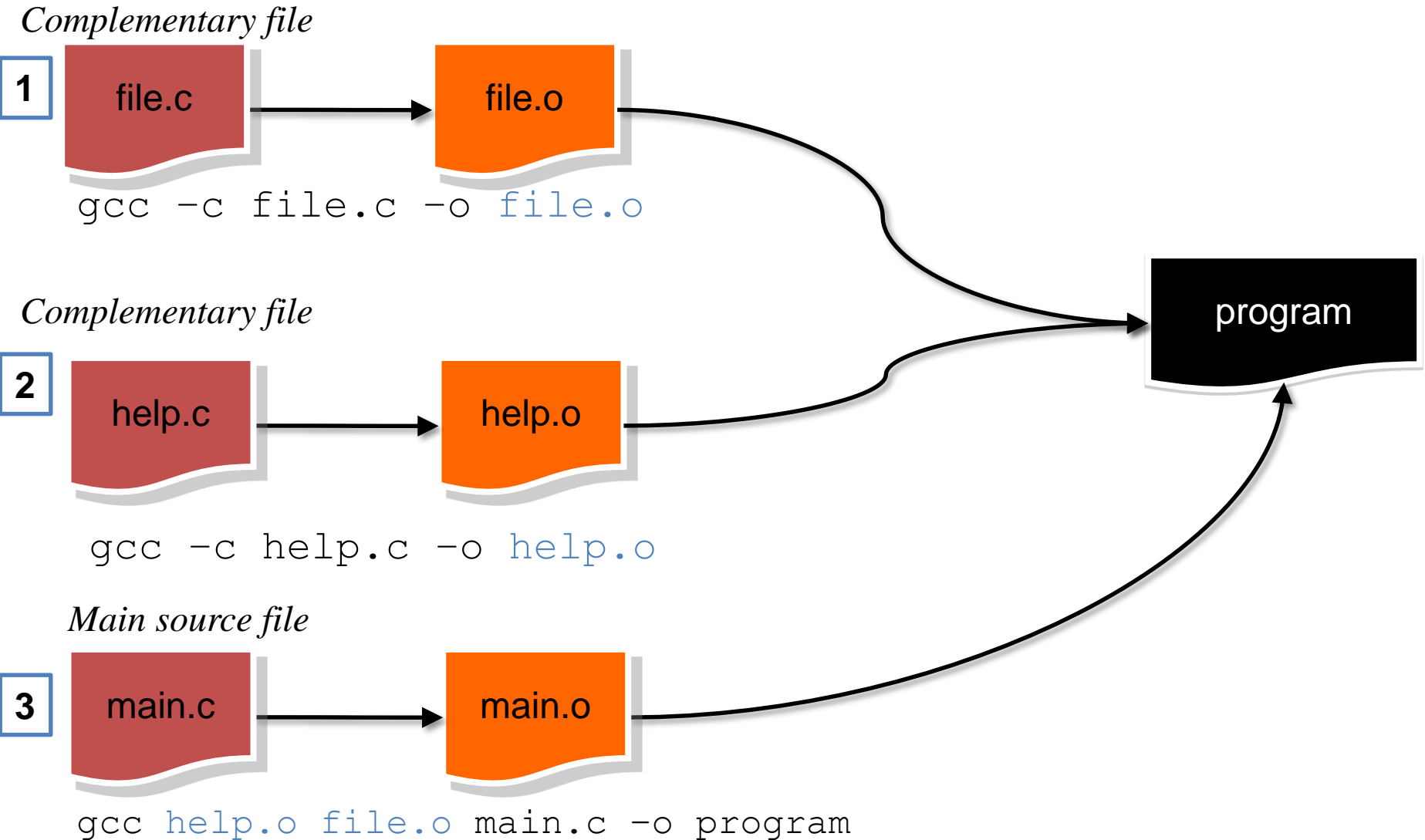
```
gcc hello.c -o program
```

- To run your program from command line (`.` indicates that the program is in the current directory)

```
./program
```

- For more details, refer to the slides

Compilation Example



Compilation Example

```
gcc -c file.c -o file.o  
gcc -c help.c -o help.o  
gcc help.o file.o main.c -o program
```



Program to execute these
commands in the right order:
'make'
It needs information about the
code which is given in form of
a Makefile.

Text editors

- To edit/write your code, you can use any text editor.
 - *gedit*, *SciTe*, *sublime*
 - Otherwise *Emacs* or *vi* for the adventurous
 - (more powerful, but less intuitive user-interface)

Reminder

- Where do I get necessary information?
 - Lecture notes
 - man pages
 - Press Tab to complete filenames
 - Web:
 - <http://cplusplus.com/reference/clibrary/>

Enum and typedef example

```
#include <stdio.h>
```

```
typedef enum
```

```
{  
    Sunday,  
    Monday,  
    Tuesday,  
    Wednesday,  
    Thursday,  
    Friday,  
    Saturday  
} WeekDays;
```

```
int main()
```

```
{  
    WeekDays day = Saturday;  
    if (day == Sunday || day == Saturday)  
    {  
        printf("weekend\n");  
    }  
    else  
    {  
        printf("work day\n");  
    }  
    printf(" It is day %d of the week\n",day+1);  
    return 0;  
}
```

Reminder

- Hint
 - for bonus questions don't spend too much time
- Take notes
- Understand the concepts

Feedback for lab 1

In order to improve the labs, help us by filling out the feedback form on moodle

Thanks!

Questions and help

- Send questions via e-mail:
 - sis-ta@groupes.epfl.ch
- Do not hesitate to ask questions!