

CS 241

Data Organization

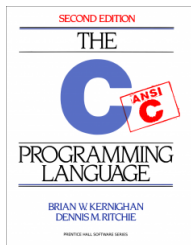
Hello World – in Linux

Brooke Chenoweth

University of New Mexico

Fall 2014

Read Kernighan & Richie



- 1.1 Getting Started
- 1.2 Variables and Arithmetic Expressions
- 1.3 The For Statement
- 1.4 Symbolic Constants

Hello, World: A Program in C

```
1 #include <stdio.h>
2 int main(void)
3 {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

{ } Curly brackets are used to group lines of code into blocks. Lines 3 through 6 make the body of **“main”**.

() Parenthesis are used for function parameters. In line 2, the parameter list for **“main”** is **“void”**.

Secure Shell to *cs.unm.edu

- PuTTY: free implementation of Telnet and SSH (Secure Shell network protocol) for Windows
- Mac OS X comes with its own implementation of OpenSSH, so you don't need to install third-party software.
- Ditto Linux
- **moons.cs.unm.edu**
trucks.cs.unm.edu

Set Up PuTTY – Session Options

- Open Putty.
- Set Host and Protocol
 - Set Host Name to **moons.cs.unm.edu** or **trucks.cs.unm.edu**
 - Set Connection Type to SSH

Set Up PuTTY – Window Options

- Set Window options
 - May want to increase Rows from the default
 - When window is resized: Change the number of rows and columns
 - May want to increase Lines of scrollback

Set up Putty – Translation

- Set Received data character set to **UTF-8**

The Linux Host sends a binary signal to PuTTY. PuTTY needs to be told which standard to use to translate that signal. Most standards use the same codes for the upper and lower case 26 letters of the 26 English alphabet.

“Special” characters, (such as ç, ñ, ©, π) often have different codes in different standards.

Set Up PuTTY – Save and Connect

- Name and save your configuration
- You can also customize other parts of the configuration:
 - Default User Name
 - Font Size
 - Foreground Color
 - Background Color
 - etc.
- Open the connection.
- In the future, you can load your saved configuration and avoid all the setup the next time.

Open the connection

- You will see a command prompt with your user name and the machine name, starting in your home directory.
- bash shell: Default Linux shell environment on cs machines
- Workflow Suggestion: Dual PuTTY
Open two PuTTY windows: one for editing and the other to compile and run.

Aside: Unix Shells

- A **Unix shell** is a command-line interpreter that provides a traditional user interface for the Unix operating system and for Unix-like systems (i.e. Linux).
- The most influential Unix shells:
 - **C shell** syntax modeled after the C programming language
 - **Bourne shell** early de facto standard
 - **bash** (Bourne-Again SHell): Superset of Bourne Shell functionality. Default interactive shell for users on most GNU/Linux and Mac OS X systems.

Aside: Poking Around Your Home Directory

`pwd` Linux command that returns the absolute path of the current directory.

`ls` Linux command that lists all files in the current directory.

`ls -F` The F is a command line option that tells ls append a "*" to the end of every executable file, a "/" to the end of every folder, and an "@" to end of every symbolic link

Create a Working Directory

`mkdir name` Linux command that creates a directory with the given name. Returns an error message if a file already exists with that name.

`cd name` Linux command that looks in the *current directory* for a directory of the given name. If the given directory is found, then the current directory is changed the given directory.

If the **cd** command is used without specifying a name, then the current directory is changed to the users *home directory*.

Open a Text Editor: emacs

- `emacs hello.c`
- `hello.c`: File name passed to emacs
- If this file does not exist in the current directory, then a new empty file is created with the given name.
- C source code should be given a file name that ends with `.c`

Aside: Text Editors

- emacs
 - <http://www.gnu.org/software/emacs/>
 - Contains multitudes (including a vi emulator!)
 - Key chords give your pinkie a workout.
- vi
 - <http://www.vim.org/>
 - Much smaller than emacs
 - Editing modes can be confusing at first
- Try them and see which you prefer.
- Both run on about every OS, so if you have network problems, edit locally and use SFTP.

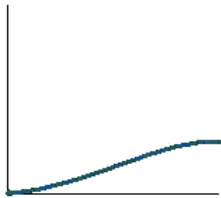
Text Editor Learning Curve

Classical learning curves for some common editors

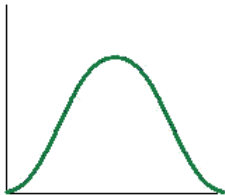
Notepad



Pico



Visual Studio



vi



emacs



Enter C Source Code

```
#include <stdio.h>
int main()
{
    printf("Hello, World!\n");
    return 0;
}
```


Save file, Compile, and Run

- **C-x C-s** saves file in emacs
- **C-x C-c** exits emacs and returns to command line
 - Alternatively, use **C-z** to suspend emacs, and type **fg** to resume.
- **gcc hello.c** Runs a C compiler program with hello.c as input.

If gcc compiles hello.c without errors, then gcc will produce a.out: an executable file containing machine code.
- **./a.out** runs the program. (The “./” tells Linux to look in the current directory for a.out)

Syntax Error

```
int main()  
{  
    printf("Hello, World!\n");  
    return 0;  
}
```

```
bchenoweth@prospero: /cs241work$ gcc  
hello.c  
hello.c: In function main:  
hello.c:3:3: warning: incompatible  
implicit declaration of built-in function  
printf
```

Look at What We Created

- `pwd` Displays absolute path of current directory (Where am I?)
- `ls -l` The `l` option of `ls` specifies listing files in the long listing format.
- Access permissions: *details later*
 - owner
 - group
 - File size in bytes.

The executable is large because we did not compile with a **dynamically linked library**. Thus, `a.out` contains all code for `printf`.

View File Contents: more

more filename

more is a filter for paging through text one screenful at a time. This version is especially primitive. Users should realize that **less** provides more emulation and extensive enhancements. Sometimes your programs output is too long to fit on the screen. (This is especially true when debugging.)

Using more

Redirect your output to a file, `>`, then use `more` (or `less`).

```
./a.out > longOutput.txt  
more longOutput
```

Alternatively, redirect output directly to `more` with a pipe. (Disadvantage is output isn't saved this way.)

```
./a.out | more
```

Hidden Files: `ls -a`

The **a** option of **ls** specifies to include hidden files. File names that start with . (period) are hidden files in Linux:

- . is the current directory.

- .. is the parent directory.

cd .. will change the current directory to the parent directory.

-F displays / after directorys and * after executables.

Linux Access Permissions

- $\underbrace{rwx}_{\text{User}} \underbrace{rwx}_{\text{Group}} \underbrace{rwx}_{\text{Other}}$

The initial “-” will be a “d” if the file is a directory.

Only executable files should have execute permission:

- machine code
- shell scripts
- directories

In Linux, a file has nine independent permission properties:

- read
- write
- execute

for each of three types of users:

- user
- group
- others

Changing access permissions with chmod

chmod *<who><+|-><permission>file*

chmod o-x a.out Remove execute permission from others.

chmod a-w a.out Remove write permission from all users.

chmod a+r a.out Add read permission to all users.

bash Environment Variable: \$PATH

- **which**: Linux command that searches your shells **\$path** variable for the given argument.
 - Returns the absolute path or “Command not Found”.
- **echo \$PATH** to view \$PATH
 - directories delimited by :
- By default, . (the current directory) is not included in \$PATH.

.bash_profile Configuration File

- When you login to a bash shell, either sitting at the machine or remotely via ssh, bash looks in your home directory for the hidden file **.bash_profile**. If this file exists, then bash will automatically execute it to configure your shell before showing you the initial command prompt when you log in.
- You can create and edit .bash_profile, *but be careful*.

.bash_profile Example

```
export PATH=$PATH:.  
alias ls="ls -F"
```

The first line adds the current directory (.) to the end of the path.

The second line creates an alias so that whenever you enter `ls` as a shell command, bash will replace it with `ls -F`.

Note: the current directory at the start of `$PATH` is considered a security risk. Why?

Linux Change Directory: cd

- `cd name` Looks in the **current directory** for a directory of the given name. If the given directory is found, then the current directory is changed the given directory.
- `cd` Sets the current directory to the users **home directory**.
- `cd .` Does nothing (why?)
- `cd ..` Sets the current directory to the parent directory.
- `cd ~user` Sets the current directory to the home directory of the specified user.

Manual Pages: man command

<i>man command</i>	displays manual pages on the specified command. Usually the man pages do not fit inside the window.
<i>spacebar</i>	moves down one page.
<i>enter or down arrow</i>	moves down one line.
<i>up arrow</i>	moves up one line.
<i>q</i>	quit man pages.

UNM Labs: Secure Telnet Connect

- Start → All Programs → Secure Telnet and FTP → Telnet

Host Name moons.cs.unm.edu or
trucks.cs.unm.edu

User Name Your cs.unm.edu user name.

- Save public key of host
- When done, use the Linux command: **exit**

UNM Labs: Moving files: SFTP

- Start → All Programs → Secure Telnet and FTP → FTP

Host Name moons.cs.unm.edu or
trucks.cs.unm.edu

User Name Your cs.unm.edu user name.

- Local file system on left, remote system on right
- Click and Drag files between systems

How Do I Connect From My Machine?

Question: At Home, I have a computer running \triangle OS version \square . How do I ssh and sftp?

Answer:

- Google \triangle \square SFTP
- UNM Information Technologies
- CS Support Group
cssupport@cs.unm.edu
Help Desk - 277-3527
FEC 30718

Handy terminal tricks

<tab> Auto complete a file name if enough letters have been typed to identify that file.
For example, if the only file in the current directory that starts with **gr** is **graphParser.c**, then **gr<tab>** will auto complete.

history Displays a history of the commands entered.
↑ Walks back through the command history.
Pressing **<enter>** on a selected command will repeat that command.

!x Repeats the most recent command starting with the sub-string x:

gcc graphParser.c

!g