CS 241
Data Organization using C

Hello World – in Linux

Instructor: Joel Castellanos
e-mail: joel@unm.edu

Web: http://cs.unm.edu/~joel/
Office: Farris Engineering Center
(FEC bldg. #119)
Room 2110

% gcc helloWorld.c

Read: Kernighan & Ritchie

- Due Thursday, Aug 22
  1.1: Getting Started
  1.2: Variables and Arithmetic Expressions
  1.3: The For Statement
  1.4: Symbolic Constants
CS Building & Website: https://www.cs.unm.edu/

Farris Engineering Center
- CS Faculty offices
- CS Department office
- CS Help and Support
- CS Labs
- CS Tutors

CS Computer Account  https://www.cs.unm.edu/

Walk-in assistance in FEC 3550
Hello World: A Program in C

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

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”.

--- INSERT ---

Getting Started

After getting an CS account

You will get a new email and storage account with CS, which is different from your UMN NetID username and password.

On the third floor of Farish, there is a computer lab in room 308, just to the right of the elevator. It has a HP LaserJet, at queue name **netid** and a color LaserJet at **color**.

There is also a poster printer that prints at a massive 42 inches wide. Talk to the systems folks first so they can give you the right settings and queue name—some of the defaults can be non-obvious. It’s located in the Systems room, PDC 367, just to the right of the elevator as you exit.

For more information and printers, check out the Printer page.

Working Remotely

There are three groups of computers which you can log on (osts) into remotely. They are: shuttle.cs.unm.edu, moons.cs.unm.edu, giga.cs.unm.edu

For Windows, use putty to connect to these systems. Linux and Mac OS X have ssh built in to the command line. You can also use Submission remotely if you have configured it.

Wireless networks
Step 1A: Secure Shell to *.cs.unm.edu

- **PuTTY**: free implementation of Telnet and SSH (Secure Shell network protocol) for Windows 95, 98, ME, NT, 2000, XP and Vista on Intel x86 platforms.

- Mac OS X comes with its own implementation of OpenSSH, so you don't need to install third-party software. From Macintosh HD and go to the **Applications** folder, then **Utilities** from within that, select **Terminal**.

- moons.cs.unm.edu

- shuttle.cs.unm.edu

Step 1B: PuTTY: Set Host & Protocol

- **Open PuTTY**

  ![PuTTY configuration screen](image)

  **Host:** moons.cs.unm.edu or trucks.cs.unm.edu

  Sometimes, one of the servers is down. If one of these does not work, then try the other.
Step 1C: PuTTY: Set Window Options

May want to increase **Rows** from the default 24.

When Window is resized: Change the number of rows and columns.

Lines of scrollback

Step 1D: PuTTY: Set Translation

Window → Translation → Received data assumed to be in which character set → **UTF-8**

The Linux Host sends a binary single 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, π, ©, ±, ñ Often have different codes in different standards.

**Good:** hello.c:24: error: ‘i’ undeclared

**Bad:** hello.c:24: error: âia undeclared
Step 1E: PuTTY: Save Configuration

Give your configuration a name and click **Save**.

You can also customize other parts of the configuration:
- Default User Name,
- Font Size,
- Foreground Color,
- Background Color,
- etc.

Step 1F: Open the Connection

**bash Shell**: Default Linux shell environment on cs machines

- **Machine Name**
- **User name**
- **Command Prompt**
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 to append a ‘/’ character to the end of directories, a ‘*’ character to the end of executable files and an ‘@’ character to the end of symbolic links.
Step 2: Create a Working Directory

**mkdir name**: Linux command that creates a directory with the given name. Returns and error message if a file already exits 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 user’s *home directory*.

Step 3: Open a Text Editor: *vim*

**vim**: text editor.

**hello.c**: File name passed to vim. 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: vim and an Empty File

Before you can start typing code into vim, you must enter the vim insert command: i

blue ‘~’ characters indicate that this space is beyond the end of the file.

Column number of cursor
Line number of cursor

file name

Aside: vim Text Editor

- vim is modal — a design choice that tends to confuse new users unaware of insert-mode.
- Run in remote graphics window.
- Run in remote text-only window.
- Free, and open source – compiled executables for ***every*** platform:
  - Atari 800
  - Small hardware devices.
- Very small and simple executable.
- Tons of documentation
- Official Website: http://www.vim.org/

Tip: if network is problematic, download vim and run locally.
Then SFTP.
Aside: vim: 1 of 3

- **i** Enter insert mode
- **v** Enter visual select mode (Use normal movement keys)
- `<Esc>` Exit current mode (insert or select)
- **y** Yank selected text
- **d** Delete selected text (when in visual select mode)
- **dd** Delete line
- **x** Delete character
- **rc** Replace one character with `c`
- **p** (P) Put yanked or deleted text *after* (before) the cursor.
- **:w** Write file
- **:q** Quit
- **:q!** Quit without saving

Aside: vim: 2 of 3

- ←↑→↓ Moves cursor left, up, right or down.
- **h j l k** Moves cursor left, up, right or down.
- `$` or `<end>` Moves to next ‘\n’ character — Very useful in long (multi-line) lines: *In VIM, you cannot just click the mouse in the text to move the cursor.*
- **0** Moves cursor to beginning of line.
- **u** Undo last action. Can be stepped backward.
- **U** Undo all changes to line
- **J** Join next line to end of current line.
- **a** Append (enter insert mode, *after* the cursor).
Aside: vim: 3 of 3

:= Display current line number.
:n Jump to line number \( n \).
/x Search for \( x \), where \( x \) is a character string.
/ Repeat last search.

:set nu Display line numbers.
:set nonu Turn off display line numbers.
% Jump to a matching opening or closing parenthesis, square bracket or a curly brace: (({}))
Jump to start or end of a C-style comment: /* */.

---

Step 4: Enter C Source Code

i (turn on INSERT mode)

Notice the colors. \texttt{vim} is aware of C syntax.

When the cursor is on a bracket or parenthesis, \texttt{vim} automatically highlights its match.
Turn this off: set noshowmatch
Steps 5, 6 & 7: Exit vim, Compile & Run

Within vim:
  \texttt{i}  (turn INSERT mode)
  \texttt{<enter code>}
  \texttt{esc}  (exit INSERT mode)
  \texttt{:w}  (save file)
  \texttt{:q}  (quit vim and return to Linux command line)

\texttt{gcc}: Runs a C compiler program with the source file \texttt{hello.c} as input.
If \texttt{gcc} compiles \texttt{hello.c} without errors, then \texttt{gcc} will produce \texttt{a.out}: an executable file containing machine code.

./\texttt{a.out}: runs the program.
./ tells Linux to look in the current directory for \texttt{a.out}.

Syntax Error

\texttt{oberon 72 \% gcc hello.c}
\texttt{hello.c: In function 'main':}
\texttt{hello.c:2: warning: incompatible implicit declaration of built-in function 'printf'}
\texttt{oberon 73 \%}
A Look at What We Created

The `ls` option of `ls` specifies listing files in the long listing format.

Access permissions: details later

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`

DESCRIPTION

`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 program’s output is too long to fit on the screen.
This is especially true when debugging.
Direct your output to a file, `>`, then use `more` (or `less`).
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 directories and `*` after executables.

```bash
moons.cs.unm.edu - PuTTY
setebos 158 % ls -F -l -a
total 28
drwxr-xr-x 2 joel ssg 4096 2010-01-27 09:46 ./
drwxrwxrwx 62 joel ssg 8192 2010-01-27 09:46 ../
-rwxr-xr-x 1 joel ssg 10931 2010-01-27 09:46 a.out*
-rwxr--r-- 1 joel ssg 71 2010-01-27 09:45 hello.c
setebos 159 %
```

Linux Access Permissions: `chmod`

In Linux, a file has nine independent permission properties: *read*, *write*, and *execute* for each of three types of users: *user*, *group*, *others*.

- `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 shell's `$path` variable for the given argument.
- Returns the absolute path or "Command not Found".

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.
- You can create and edit `.bash_profile`, but be careful.
- For example, `vim .bash_profile`
  ```bash
  export PATH=$PATH:
  alias ls="ls -F"
  ```
  Note: the current directory at the start of $PATH is considered a security risk. Why?

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`.
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 user’s *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`

`man command` displays manual pages on the specified command.

Usually the man pages do not fit inside the window.

<spacebar> moves down one page.

<enter> or <down arrow> moves down one line.

<up arrow> moves up one line.

q  quit man pages.
Start → All Programs → Secure Telnet and FTP → Telnet

- moon.cs.unm.edu
- trucks.cs.unm.edu

Your cs.unm.edu user name.

Save public key of host
Connected to moon.cs.unm.edu

When done, use the Linux command: exit
Start → All Programs → Secure Telnet and FTP → FTP

- moons.cs.unm.edu
- shuttle.cs.unm.edu

Your cs.unm.edu user name.

Local File System
Remote File System

Click & Drag
How Do I Connect From My Machine?

**Question:** “At Home, I have a computer running \( OS \) version \( \square \). How do I telnet and ftp”?

**Answer:**

![Google SFTP](image)

CS Support Group

[cssupport@cs.unm.edu](mailto:cssupport@cs.unm.edu)

Help Desk - 277-3527

ECE 214A

---

**Workflow Suggestion: Duel PuTTY**

Open two PuTTY window: one for editing and the other for compile and run.
Power Linux

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;tab&gt;</code></td>
<td>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 <code>gr</code> is <code>graphParser.c</code>, then <code>gr &lt;tab&gt;</code> will auto complete.</td>
</tr>
<tr>
<td><code>history</code></td>
<td>Displays a history of the commands entered.</td>
</tr>
<tr>
<td>↑</td>
<td>Walks back through the command history. Pressing <code>&lt;enter&gt;</code> on a selected command will repeat that command.</td>
</tr>
<tr>
<td><code>!x</code></td>
<td>Repeats the most recent command starting with the sub-string <code>x</code>: <code>gcc graphParser.c</code> !g</td>
</tr>
</tbody>
</table>