

GVIM and Remote Linux Work Environment

CS 241

Data Organization using C

Instructor: **Joel Castellanos**

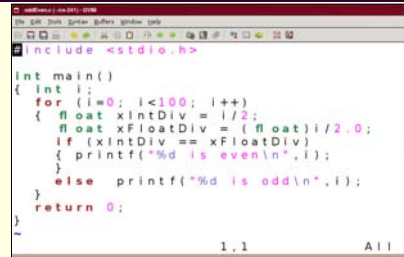
e-mail: joel@unm.edu

Web: <http://cs.unm.edu/~joel/>

Office: Farris Engineering
Center (FEC) room 321

Lab Instructor: **Dongye Meng**

e-mail: dymeng@cs.unm.edu



```
int main()
{
    int i;
    for (i=0; i<100; i++)
    {
        float xIntDiv = i/2;
        float xFloatDiv = (float)i/2.0;
        if (xIntDiv == xFloatDiv)
        {
            printf("%d is even\n", i);
        }
        else printf("%d is odd\n", i);
    }
    return 0;
}
```

3/6/2009

Turn-in policy

- All code files must be ASCII text with the file extension: `.c`
- All header files must be ASCII text with the file extension: `.h`
- All source files must have your first and last name as part of the file name:
JoelCastellanos-postfixCalculator.c
- Do not turn in executable files.
- If your turn-in includes more than 2 files, create a zip archive.

2

Connecting 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

3

PuTTY settings and commands



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

```
linkedList.c:24: error: 'i' undeclared  
linkedList.c:24: error: 'i' undeclared
```

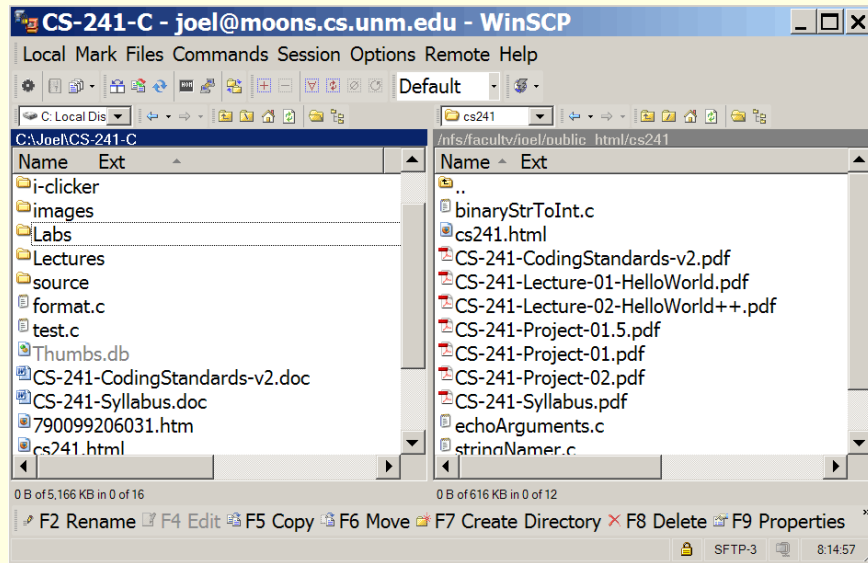
From PuTTY

shift – Insert: paste text in Windows clipboard into PuTTY (in gvin, you must first be in insert mode).

ctrl – Insert: copy selected text to Windows clipboard.

4

WinSCP: Moving Files to/from cs.unm.edu



5

Linux `ls` command



- `ls` List files in current directory
- `ls` directory List files in the specified directory
- `ls -l` List files with details such as size and date of creation.
- `ls -F` List files appending a '*' character to executable file names and a '/' character to directory names
- `ls -a` List files including hidden files (files whose name starts with '.').

6

Basic Linux Commands – Part 1 of 2



```
mkdir directoryName // Make directory  
cd directoryName // Change directory to  
directory name  
cd .. // Change directory to one level up.  
cd ~username // Search directories for home  
directory of specified user.  
cd // Change directory to your home directory.  
pwd // Print full path name of current directory.
```

7

Basic Linux Commands – Part 2 of 2



```
gcc file.c // compile c source code. If the code  
successfully compiles, it produces  
an executable file: a.out.  
ctrl - c //Exit infinite loop  
rm filename //removes (deletes) the given file.  
rm foo* //removes all files with filenames that  
start with "foo".  
rm foo * //removes foo and all files in the  
directory.  
8 rmdir directoryname //removes the given directory.
```

.cshrc



```
gvim .cshrc
```

`set path = (lots of stuff already there)` When you type a command at the command prompt, Linux searches for the command in each directory, in order, in your path. Add "\ ." to **end** of list. There is a security risk of placing it at the start of the list.

```
alias ls 'ls -F'
```

```
source .cshrc
```

9

GVIM Text Editor



- Acronym for "GNU *Vi* *IM*proved"
- vi (around since 1976) with syntax colors and menus.
- gvim 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.
- Run and edit locally, then ftp for gcc and run.
- Free, and open source – compiled executables for ***every*** platform (including the Atari 800)
- Tons of documentation everywhere
- Website: <http://www.vim.org/index.php>

10

Colors: Syntax → Automatic Syntax → Show filetypes → C



```
oddEven.c (-cs-241) - GVIM
File Edit Tools Syntax Buffers Window Help
int main(int argc, char *argv[])
{ int i=0;
  int number = 0;

  if (argc != 2)
  { printf("Error** strToInt takes exactly one arg
    return -1;
  }

  char *inputString = argv[1];
  while (inputString[i])
  { char c = inputString[i];

    if (c < '0' || c > '1')
```

11

gvim Commands: Part 1 of 2



gvim bits.c	Start with filename: bits.c.
i	Enter insert mode
<Esc>	Exit insert mode
:w	Write file
:q	Quit
:q!	Quit without saving
x	Delete character
rx	Replace with x
dd	Delete line
u	Undo
j	Join lines

12

gvim Commands: Part 2 of 2



`/x` Search for *x*, where *x* is a string.
`/` Repeat last search.
`:=` Display line number.
`:x` Jump to line number *x*.
`:set nu` Display line numbers.
`:set nonu` Turn off display line numbers.
`:syn on` Turn on syntax highlighting (colors and bold come through on a PuTTY connection).