

EC330 Discussion 2

Shane McCormack

Overview

- Command Line (Bash) Tutorial
 - Basic Commands
 - Navigation
 - File Manipulation
 - Remote Access
 - C++ Compilation
- C++ Tutorial
 - C vs C++ Strings
 - HW 1 Q4: Getting Arguments into Command Line with argv&argc

Bash Tutorial: Installation

- Mac/Linux Users: Open Terminal, Preinstalled
- Windows:
 - 3rd Party programs:
 - MobaXTerm (Home Edition): <http://mobaxterm.mobatek.net/download.html>
 - Cygwin: <https://cygwin.com/install.html>
 - Microsoft's Bash on Windows 10:
 - (Win Key) + type "use developer features"; left click first result
 - Enable Developer Mode and accept prompts
 - (Win Key) + type "bash"; left click first result
 - Bash window will open, type yes if prompted to.
 - If "legacy mode" is enabled, right click top bar of window->click "properties"->uncheck "legacy mode"->rerun bash
 - Wait for download, then create your username and password for your Ubuntu VM user
 - When typing passwords, characters won't show. This is okay! Type as normal, this is a privacy measure
 - (Win Key) + type "Ubuntu" and click "Bash on Ubuntu on Windows"
 - This is how you'll open your bash prompt from now on!

Bash Tutorial: Navigation

- Useful commands:
 - ls
 - “LiSt” shows user’s current directory
 - cd {absolute or relative path to new directory}
 - “Change Directory” Makes user’s current directory the given directory.
 - If no directory is given, will navigate to current user’s “home” directory
 - Home directories, /home/{username}, are directories which are private to the owning user. Also displayed as ~ for short
 - pwd
 - “Print Working Directory”, Outputs absolute path to user’s current directory

Bash Tutorial: File Manipulation

- Useful commands:
 - `cp {path to file to be copied} {path to where file is copied}`
 - “CoPy” Copies first file specified into second location/file specified
 - Can use absolute or relative paths
 - Can specify different file name in second argument from first
 - Will assume same name if no name is given in second argument
 - `mv {path to file to be moved} {path to where file is moved}`
 - “MoVe” Moves file supplied in first argument to path specified in second argument
 - Basically same as `cp`, but deletes the original file. `cp` rules above apply
 - Can be used to rename files
 - `zip {.zip file name} {file1 for compressed} {file2 for compression} ...`
 - Compresses multiple files to one .zip file
 - `-r` flag after “zip” for compressing directories as well as files
 - `unzip {.zip file name}`
 - Decompresses a .zip file

Bash Tutorial: Remote Access

- Useful commands:
 - `ssh {username}@{URL/IP}`
 - “Secure SHell”: Logs into server at specified IP/URL as specified user.
 - `scp {path to local file to be copied} {username}@{url/IP}:{path where file is copied to}`
 - “Secure CoPy”
 - Copies local file to remote location
 - `scp {username}@{URL/IP}:{path to where file to be copied is}`
 - Copies remote file to local location
 - `sftp {username}@{URL/IP}`
 - Opens bash-like prompt with both local and remote filesystems available
 - `get {remote filename}` downloads specified file to local machine
 - `put {local filename}` uploads specified file to remote machine
 - `bye` exits sftp
 - More detailed sftp tutorial here: <https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server>

Compiling Code

- On lab computers & via eng-grid
- Use `module load gcc` to load up correct version of GCC compiler
- Use `g++ -o desired_program_name -std=c++11 file1.cpp file2.cpp file3.cpp ...` to compile – this is how graders will compile your code

C++ Tutorial: Strings

- C-Style Strings
 - Array of chars (`char myCString[] = "I'm a C-Style String!";`)
 - Automatically ends in a null byte (`"\0"`) which indicates the end of the string
 - Because of this, the length of C-style strings is 1 more than expected
 - "Built-in"
- C++ Strings
 - String object (`string myCplusplusString = "I'm a C++ String!";`)
 - Included in `std::string`

C++ Tutorial: Command Line Arguments

- argc: “Argument Count”
 - Value = number of arguments passed at the command line
 - Name of program (“a.out”) is the first argument
- argv: “Argument Vector”
 - Array of argument values
 - argv[0] = name of program
- Usage tutorial: <http://www.cplusplus.com/articles/DEN36Up4/>