TOOLS OF THE TRADE

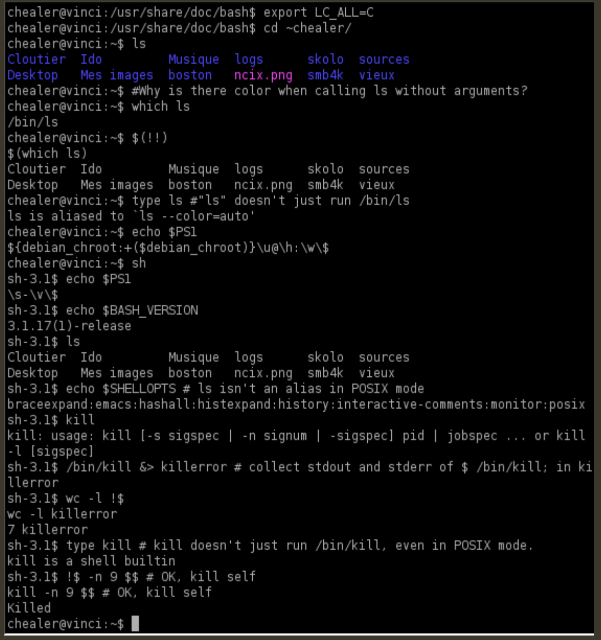
Today

- Intro to Command Line
- Intro to Git
- Lab - Recommended Setup

COMMAND LINE

What is Linux

- Popular, open-sourced computer software environment that competes with Microsoft and the Apple Macintosh. It has four major parts
  - Kernel
  - Supplied Programs
  - The Shell
  - X – KDE, GNOME



```
chealer@vinci:/usr/share/doc/bash$ export LC_ALL=C
chealer@vinci:/usr/share/doc/bash$ cd ~chealer/
chealer@vinci:~$ ls
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
chealer@vinci:~$ #Why is there color when calling ls without arguments?
chealer@vinci:~$ which ls
/bin/ls
chealer@vinci:~$ $(!!)
$(which ls)
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
chealer@vinci:~$ type ls #"ls" doesn't just run /bin/ls
ls is aliased to `ls --color=auto'
chealer@vinci:~$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@\h:\w\$
chealer@vinci:~$ sh
sh-3.1$ echo $PS1
\s-\v\$
sh-3.1$ echo $BASH_VERSION
3.1.17(1)-release
sh-3.1$ ls
Cloutier  Ido         Musique  logs      skolo  sources
Desktop   Mes images  boston   ncix.png  smb4k  vieux
sh-3.1$ echo $SHELLOPTS # ls isn't an alias in POSIX mode
braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor:posix
sh-3.1$ kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]
sh-3.1$ /bin/kill &> killerror # collect stdout and stderr of $ /bin/kill; in ki
llerror
sh-3.1$ wc -l !$
wc -l killerror
7 killerror
sh-3.1$ type kill # kill doesn't just run /bin/kill, even in POSIX mode.
kill is a shell builtin
sh-3.1$ !$ -n 9 $$ # OK, kill self
kill -n 9 $$ # OK, kill self
Killed
chealer@vinci:~$
```

What is a command line?

A CLI (command line interface) is a **user interface to a computer's operating system** or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth.

Advantages of using a command line

- Automating Task & Scheduling
- Standardization
- Efficiency
- Scripting

What is a command?

Program name + options & arguments

Examples:

```
$ wc –l myfile
$ grep -a ’full stack' example.txt
$ echo "Hello world" > helloworld.txt
$ ls | head -3 *** <- pipe is useful
```

The Shell - Bash

- Possible things – File Location, File viewing, Directory Operation, File Comparison, Network Connection, Email, Web Browsing, etc.
- cd – change directory
- ls – list files
- mkdir – create directory
- echo – print
- grep – search, Etc
- Info, --help, man

Bash- ssh

- SSH (Secure Shell)
    - Access textual shell of remote machine

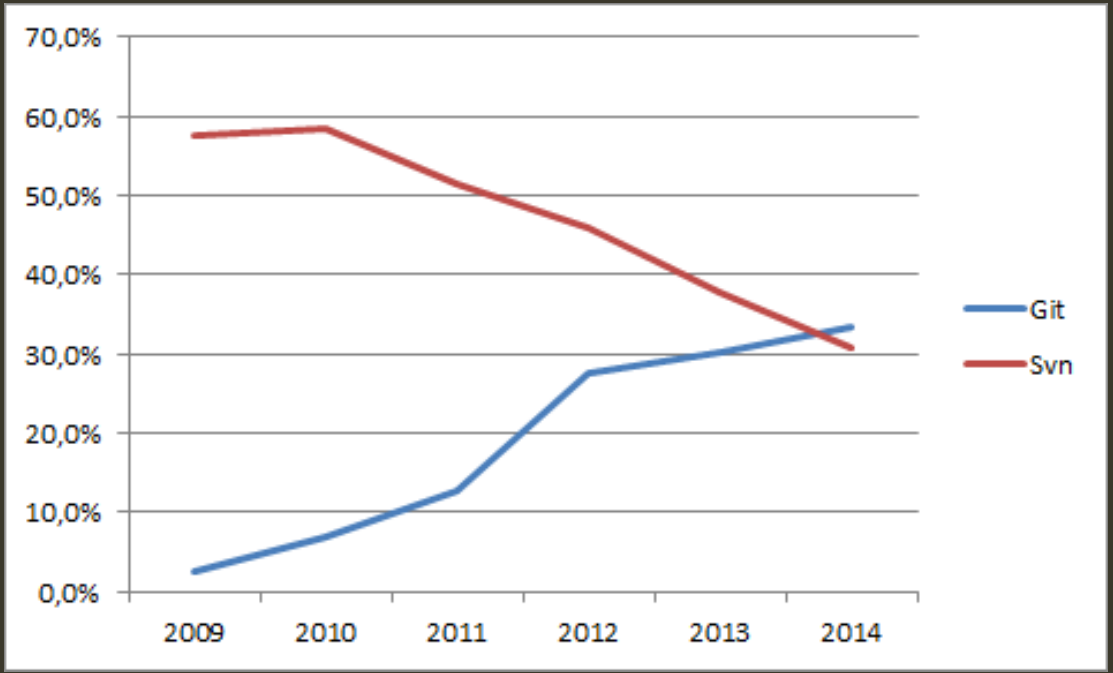
BASH - SFTP

- SFTP = SSH File Transfer Protocol
- Transfer files between two or more machines

INTRO TO GIT

What is version control?

- Version control manages changes to the source code overtime.
- Keeps tracks of all modifications to code in special database.
- Prevent concurrent work from conflicting
- Incompatibility of work should be discovered and solved
- Also known as source doe management tools



What is Git?

- Developed in 2005 by Linus Torvalds, famous creator of the Linux Operating System Kernel
- Works well on IDE
- Not fooled by name, focus on the content
- Secure hashing algorithm called SHA1
- Poweful but has steep learning curve
- A distributed Version Control Systems
- Free and Open Source
    - Complete Long-term History
    - Branching and Merging
    - Traceability - being able to annotate each change and trace changes
    - REALLY shines when you are decentralized.

Intro to Git

- git init – creates git repository
- git clone – copies existing git repository
- git add – adds a change in the working directory to the staging area (as a buffer btw directory and history)
- git commit – commit the staged snapshot to the project history
- git status – display the state of the working directory and the staging area
- git log – display committed snapshots
- git branch – represents an independent line of development, like creating new project history
- git push, git pull, git merge, git reset, git revert, git rebase

Advice

- Git is really useful during your final project
- But it can also be painful
- Make sure you and your teamates’ understanding of Git workflow are on the same page

RECOMMENDED SETUP

Necessary Software

* In today’s lab, we will install and configure the necessary software

- Text Editor
- Git
- Python 3
- Bash
- Modern Browser

Text Editor: Quick Note

Choose wisely, it is an essential part of your toolset Text Editors

- Examples: Sublime Text, Atom, Notepad, nano, etc

IDEs - Integrated Development Environments

- Language Independent: Eclipse, Netbeans, Komodo, etc
- Language Specific: Pycharm, RubyMine, VisualStudio, etc
- Not always free
- Full Featured – many that you may not use

Recommended Setup

OSX/Linux	**Windows**
Package Manager: Homebrew and Cask	Package Manager: Chocolatey
Bash: Terminal or Iterm2	Bash: Linux VM via Vagrant
Text Editor: Atom or Sublime Text 3	Text Editor: Atom or Sublime Text 3
Version Control: Git	Version Control: Git