

cs 143A

Principles of Operating Systems

Instructor : Prof. Anton Burtsev (aburtsev@uci.edu)

Assistants :

- Junjie Shen
junjies1@uci.edu
- Vikram Narayanan
narayav1@uci.edu
- Biswadip Maity
(Deep)
Email : maityb@uci.edu

Please include “143A” in the subject line

Today's agenda

- Accessing Andromeda Shell using SSH
 - Understanding a few Shell commands
 - Introduction to git
- If students are already familiar with shell commands/git, we can move on to:
- Downloading the XV6 source code

All important content will be uploaded in the course page.
Taking notes is not required.

Following hands-on is encouraged during the session

I have ICS account

1. Find out your student ID
(Ex: 66541280)
2. Evaluate serverNumber =
(studentIDNumber mod 74) +1

Ex: 66541280 => 37
3. Your server name is :
andromeda-<serverNumber>
Ex: andromeda-37.ics.uci.edu
4. Make a note of it.

I do not have ICS account

- Each member of the ICS community receives an ICS email address of the following form:
username@ics.uci.edu
- ICS students can visit Computing Support (ICS 364, bldg. 302 on the campus map) for your ICS computer and email accounts. Please take your ID for activation.
 - Other students– Please talk to me after class for alternative options

SSH client

SSH into andromeda-XX.ics.uci.edu

Ex : SSH maityb@andromeda-XX.ics.uci.edu
Enter your ICS account password when prompted

I have ICS account

I do not have ICS account (SSH required for next class)



Supports bash.
<Check if SSH works>

Putty/
Other SSH Client

SSH from terminal

SSH from terminal



Use bash terminal

- Download Cygwin/ alternatives for today
- Partner up

Use terminal

Use terminal

Andromeda Command Promt

SSH into andromeda-XX.ics.uci.edu

Ex : SSH maityb@andromeda-XX.ics.uci.edu
Enter your ICS account password when prompted

Before we start

All Linux operating systems function in lower case.

While you can name files, folders and directories using upper case, the system functions in lower case. This means unless you specify -i (negate case lock), all files, folders and directories named with an upper case will not be shown.

Thus, the command

locate operatingsystems.mp3

Will not locate the file OperatingSystems.mp3

Before we start

Be very careful using the rm command.

The rm -rf/ command means remove (rm) - recursive (r) force (f) home (/).

Spelled out logically, the rm -rf/ command will delete every folder, file and directory within your Linux OS. It is the equivalent of wiping your entire hard drive clean.

<http://www.informit.com/blogs/blog.aspx?uk=The-10-Most-Important-Linux-Commands>

Before we start

/ means root

The PC user in command prompt will start from the C:\ hard drive. In Linux your CLI starting point - your root directory - is /.

/ is where all other files, folders and directories are stored within.

Before we start

Passwords are kept in the dark

This is because when you type in your password into the Linux CLI, you will see nothing.

The Linux CLI operates with the assumption that the user typing in the password knows what he/she is typing and thus, for security reasons, has no need to view it while it is input.

<http://www.informit.com/blogs/blog.aspx?uk=The-10-Most-Important-Linux-Commands>

Some useful commands

1. cd /bin
2. ls

Some useful commands

1. cd /bin
2. ls
3. ls -1

Some useful commands

1. cd /bin
2. ls
3. ls -1
4. ls -1 | grep m

Some useful commands

man

The man command - the manual command - is used to show the manual of the inputted command. Inputting the man command will show you all information about the command you are using. An example:

```
man cd
```

<http://www.informit.com/blogs/blog.aspx?uk=The-10-Most-Important-Linux-Commands>

Some useful commands

mkdir

The mkdir - make directory - command allows the user to make a new directory. An example of the mkdir command

```
cd ~  
mkdir ics143a
```

<http://www.informit.com/blogs/blog.aspx?uk=The-10-Most-Important-Linux-Commands>

Some useful commands

echo

```
echo 'Hello, world.'
```

```
echo $USER
```

```
echo $Home
```

```
echo ~
```

Some useful commands

echo with redirect

```
echo 'Hello, world.' > foo.txt
```

Or if you want to append data:

```
someCommand >> someFile.txt
```

Some useful commands

cat

View contents of a file

Ex: cat foo.txt

Other uses :

- create single or multiple files
- concatenate files
- redirect output in terminal or files.

Text Editor

vim

VIM has two modes in which you can manipulate files:

- Edit mode is the state in which the keys you bang on are actually inserted into your document
- GUI layout allows you to navigate through the document, search and replace text, copy and paste, etc.

By default when you open a file, you're placed in GUI mode.

Text Editor

vim – sample commands

- h moves the cursor one character to the left.
- j moves the cursor down one line.
- k moves the cursor up one line.
- l moves the cursor one character to the right.
- 0 moves the cursor to the beginning of the line.
- \$ moves the cursor to the end of the line.
- w move forward one word.
- b move backward one word.
- G move to the end of the file.
- gg move to the beginning of the file.
- . move to the last edit.

Search for Vim 101 for full list of commands

Text Editor

vim - exit

If in the edit mode, Hit the ESC key to get into *Normal* (command) mode then type :q and press Enter.

To quit without saving any changes, type :q! and press Enter.

Search for Vim 101 for full list of commands

Version controlling system

create a new repository

create a new directory, open it and perform a

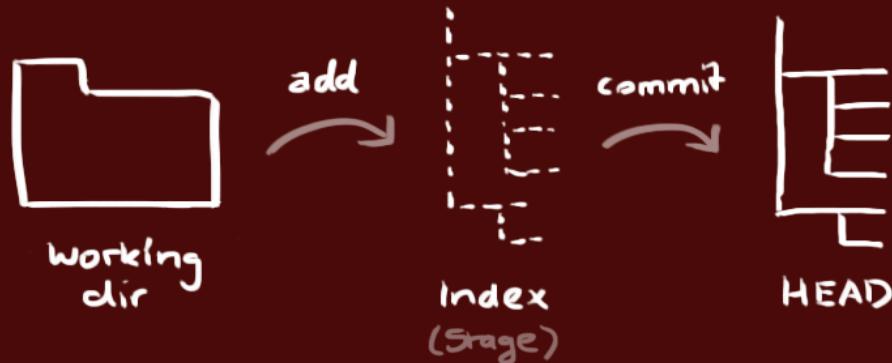
`git init`

to create a new git repository.

Source :<http://rogerdudler.github.io/git-guide/>

Version controlling system workflow

your local repository consists of three "trees" maintained by git. the first one is your **Working Directory** which holds the actual files. the second one is the **Index** which acts as a staging area and finally the **HEAD** which points to the last commit you've made.



Source :<http://rogerdudler.github.io/git-guide/>

Version controlling system

add & commit

You can propose changes (add it to the **Index**) using

```
git add <filename>
```

```
git add *
```

This is the first step in the basic git workflow. To actually commit these changes use

```
git commit -m "Commit message"
```

Now the file is committed to the **HEAD**, but not in your remote repository yet.

Source :<http://rogerdudler.github.io/git-guide/>

Version controlling system

git demo

- git init
- git status
- git log
- git add
- git commit
- git reset --hard HEAD~1
(Resets the index and working tree. Any changes to tracked files in the working tree since <commit> are discarded.)
- git reset --soft HEAD~1
(Does not touch index or the working tree, puts changes in staged area)

Thank You

Next day : Booting XV6