In this episode...

- Connecting with Andromeda.
- Linux. A survival guide.
Andromeda

• Server with Linux managed by the ICS.
• You must have an “@ics.uci.edu” account.
• Your server number is:
  \[ \text{ser\_num} = (\text{<ucinetid>\%75})+1 \]
• The server you will connect is:
  \text{andromeda-<ser\_num>.ics.uci.edu}
How to connect to it

We need a secure shell client (ssh).

- In Linux: integrated.
- In Windows: integrated (win10), or Putty.
- In Mac-OS: integrated.

ssh username@andromeda-XX.ics.uci.edu
Welcome to Linux, now what?

A couple of notes:

- `/` is the root directory, everything is under it.
- `~` is your home directory, it is an alias for `/home/yourUsername/`
- `./` is the current directory.
- `../` is the parent directory.
- Passwords are invisible.
- Case matters, “A” and “a” are different.
Some commands

```
ls [options] [dir]
```

- List information about the FILEs (the current directory by default).
  - -R: recursive.
  - -l: long format, shows info of each file
  - -a: show all files, including hidden files, those that start with a “.”
  - -h: file sizes in a nice way
Some commands

```
cd [dir]
```
Change the shell working directory.

• If no directory is given, it changes to the home directory.
Some commands

`pwd`

Print the name of the current working directory.
Some commands

```
mkdir <dir_name>
```
Create the directory `<dir_name>`, if it does not already exist.
Some commands

touch -c <filename>

Creates a new empty file.
Some commands

```bash
cp <source> <dest>
```
Copy the source file to the destination.

• Example:
```bash
cp myFile.txt ./aDirectory/newFile.txt
```
Some commands

`mv <source> <dest>`

Move or rename the source file to the destination file.

- Example moving:
  ```
  mv myFile.txt ./aDirectory/newFile.txt
  ```

- Example renaming:
  ```
  mv myFile.txt newName.txt
  ```
Some commands

```
rm [op] <filename>
<dir>
```

delete a file in the specified directory. If no directory is given, uses the current one.

- \texttt{-r}: recursive.
- \texttt{-d}: remove empty directories.
- \texttt{-i}: interactive, ask before each file.

**DO NOT TRY THIS:** \texttt{rm -rd ./}
Some commands

cat [filename]

Print the content of a file to standard output. If no file is given, then prints what comes from standard input (most of the cases the keyboard)
Some commands

```
head <filename>
```

Print the first 10 lines of the filename to standard output.

- `-n X`: prints the first $X$ lines
Some commands

```
tail <op> <filename>
```

Print the last 10 lines of the filename to standard output.

- `-n X`: prints the last $X$ lines
Some commands

grep <pattern> [file]
Search for <pattern> in the given file.

• If no file is given, grep searches recursively in the working directory.

• Using -e <pattern>, grep interprets the pattern as an extended regular expression.
Some commands

```
find -name <filename> <dir>
```

Find a file named `<filename>` in the directory `<dir>`

- There is a lot of other options.
Some commands

```bash
echo "<message>"
```

Print the `<message>` in the standard output

- `-e`: enable interpretation of backslash escapes, example:
  ```bash
  echo -e "first line\nsecond line"
  ```
Other useful tools

“>” and “<”

Redirect the standard input or output.

- Example: `ls -l > myContent.txt`
- Example: `cat < myContent.txt`
Other useful tools

">>"
Append.

• Example:
  echo "Dear diary" > log.txt
  echo "Today I..." >> log.txt
Other useful tools

| Pipe, passes the output of a command to another as input. |

- Example:
  
  `ls -R | grep "myFile"`
Editors

emacs

- C = Ctrl, M = Alt
- C-x C-f visit (open) a file
- C-n next line
- C-p previous line
- C-b back (move left)
- C-f forward (move right)
- C-a beginning of line
- C-e end of line
- C-x C-s save
- C-z suspend (minimize)
- C-x C-c close (exit)

- C-space toggle selection
- M-w copy
- C-w kill (cut)
- C-y yank (paste)
- C-/ undo
- C-s search
- C-x 2 split windows horizontally
- C-x 3 split windows vertically
- C-x 0 clear this splitting
- C-x 1 clear all splitting
- C-h getting help
Copying things

```
scp <source> <destination>:<dir>
```

Copy files from one machine to another through ssh. Easier if you run it in your local machine

Example local to remote:

```
scp ~/localFile peter@andromeda-XX.ics.uci.edu:~/remoteFile
```

Example remote to local:

```
scp peter@andromeda-XX.ics.uci.edu:~/remoteFile ~/localFile
```