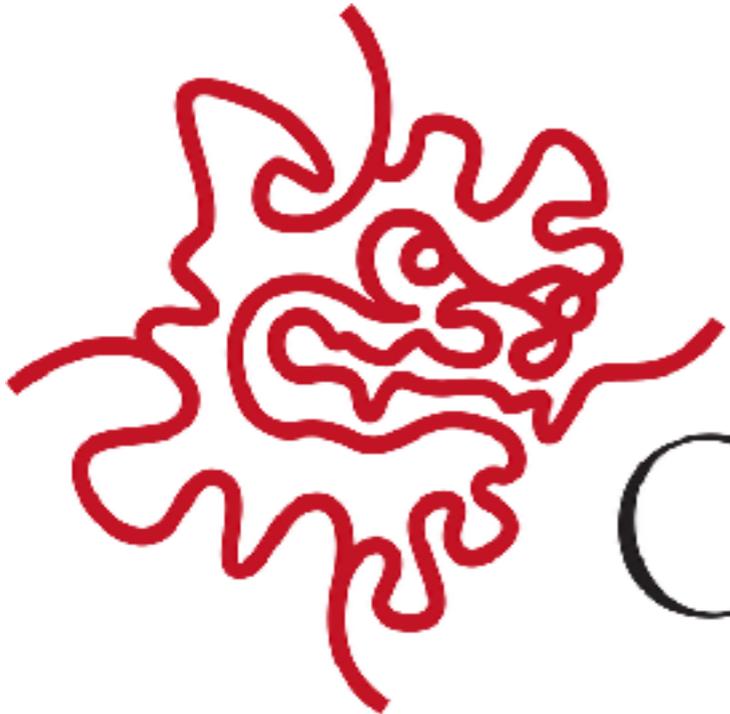




Basic Terminal

Angela White



OIST

- Navigate
- Manipulating, creating and removing files
- Permissions
- Piping
- Useful information
- Shortcuts



- **STEP 1: Open up a terminal**



- Change directory `> cd /path`
- Go back one directory (to the directory above you) `> cd ..`
- Go home `> cd ~` `> cd`
- Print working directory `> pwd`
- list directory `> ls`



- **STEP 2: Find out where you are** > `pwd`
- **STEP 3: List everything in your current working directory** > `ls`
- **STEP 4: Move into a 'Documents' or some other folder / directory** > `cd Documents`
- **STEP 5: Move back 'home'**
> `cd ~`
> `cd`



List directory > ls

- List hidden files too > ls -a
- Show permissions > ls -l
- Sort by time modified > ls -q

Show command history > history



- **STEP 6: Show all the commands you've tried in the terminal so far**

> history



Directory

- Make a directory > `mkdir directory_name`
- Remove a directory > `rmdir directory_name`

Files

- Make a file > `touch file_name`
- Remove a file > `rm file_name`



BE CAREFUL

```
> rm -rf / ← NEVER
```

- **DANGEROUS:** Recursively, forcefully remove a directory and all of its contents:

```
> rm -rf directory_name ← NEVER
```

- **MUCH SAFER** Remove a single file:

```
> rm file_name
```

- **MUCH SAFER** Remove

```
> rm -i file_name
```

```
> rm -rf -i directory_name
```



- Copy a file > `cp file_name new_name`
- Move a file > `mv old_place new_place`
- Print text from a file > `cat file_name`
- Print first few lines of file > `head file_name`
- Print last few lines of file > `tail file_name`
 - > `head -n 5 file_name`
 - > `tail -n 5 file_name`



- **STEP 7: Create a directory**
- **STEP 8: Put some new files in there remove a few of them.**
- **STEP 9: Put a few lines of random text in one of the files, move this file to a subdirectory and print out the last line in the file and the first two lines.**



- Permissions: Who can access a file or directory & how
Read: **-r** (=4) Write: **-w** (=2) Execute: **-x** (=1)
- Who? Owner of the file: **'user'**
Members of the group who own the file: **'group'**
Anyone else: **'others'**
- Change mode:

> *chmod options permissions file_name*

Symbolic: > *chmod u=rwx,g=rw,o=r file_name*

Numbers: > *chmod 764 file_name*



- Invoke root access for one command:

> `sudo command`

- Become root:

> `su`



- **STEP 10:** Change permissions of a file so that you (the owner) can read and write it, the group can read and write it and anyone else can only read it.
 - > `chmod u=rw,g=rw,o=r file_name`
 - > `chmod 662 file_name`
- **STEP 11:** Use the `su` command



- Piping “|”
Inputs output from the program on the left to the program on the right **> A | B**
> ls | head -4 | tail -1
- **>** Rewrites output to file
- **>>** Appends output to the file
- **echo** Repeats what you put in the file
> echo “Hello world” >> world
- **grep** Prints lines matching a pattern
> grep “world” world
> grep -v “moon” world



- **STEP 12:** Make a new file and use piping to transcribe a few line poem there, i.e:

*Big whirls have little whirls,
That feed on their velocity;
And little whirls have lesser whirls,
And so on to viscosity.*

- **STEP 13:** Print out the second last line of the poem.

```
> tail -n 2 poem | head -n 1
```



- > `man something` :Help manual
- > `top` :Summary of running system; system summary information; threads run by kernel. User configurable.
- > `free -m` : (sango) Free memory
- > `ps` :Lists active processes - without arguments, just processes associated with current user and terminal. > `ps aux` - processes owned by all users
- > `ps -u user` - process owned by specific user
- > `pgrep process` - to find the process id of a specific process



- Kill a process: **>kill** **>pkill** **>killall**

>kill *process_pid* Sends the TERM signal to the process which tells it to terminate

>kill -KILL *process_pid* Sends to the operating system kernel which kills the process

>kill -l All signals that are possible to send with kill

>pkill *process_name* Same as **kill** but using process name

>killall *process_name* kills every instance of a process



>du -h *file* Size of a file

>df -h Information on free space

>lspci | grep VGA

Information about your computer



- **STEP 14:** Start a process, see if you can find how its running and then kill it, all from the command line.
- **STEP 15:** Find if you have any free disk space, investigate!



- **TAB** :Autocompletes
- ***** : Means ‘all like this’
- **CTRL + C**: Terminate / interrupt process
- **CTRL + D**: Log out
- **CTRL + L**: Clears terminal
- **CTRL + A**: Moves cursor to start of line
- **CTRL + E**: Moves cursor to end of line
- Spaces are “\ ” [BACKSLASH SPACE] in the terminal



- **STEP 16:** Make some files with similar names and clear them all with one command.
- **STEP 17:** Make a file with “Are terminals useful?” and replace the text with “Terminals are useful!”. Print out to check.
- **STEP 18:** Clear and log out of your terminal



Do you have an account on Tombo / Sango?

```
> ssh username@tombo.oist.jp
```

No? Please sign up for one!

User activation request:

<https://groups.oist.jp/scs/registration-forms>



Finally!

SKILLPILLS

Terminal help session with Jan Moren
from Scientific Computing & Data
Analysis Section.

May 9-10th

Keep an eye on tida for the details!

