H3ABioNet

معهد باستور تونس
Institut Pasteur de Tunis

H3ABioNet
Pan African Bioinformatics Network for H3Africa

Introduction to Bioinformatics online course: IBT

Linux

Manipulating files

Learning Objectives

① Learn how to create and edit files

② Learn how to view files content

③ Learn basic commands to manage files and directories

④ Learn some useful wildcards

Learning Outcomes

① Be able to create and edit files

② Be able to view files content

③ Be able to manage files and directories

④ Be able to use some wildcards

Part 1

Creating new files

Basics manupulating file commands

- touch is used to create, change and modify timestamps of a file
- touch command creates an empty (zero byte) new file using this

  Structure: touch filename

- Create more than one single file

  touch filename1 filename2 filename3

touch command options

- -a: change the access time only
- -c: if the file does not exist, do not create it
- -d: update the access and modification times
- -m: change the modification time only
- -r: use the access and modification times of file
- -t: creates a file using a specified time

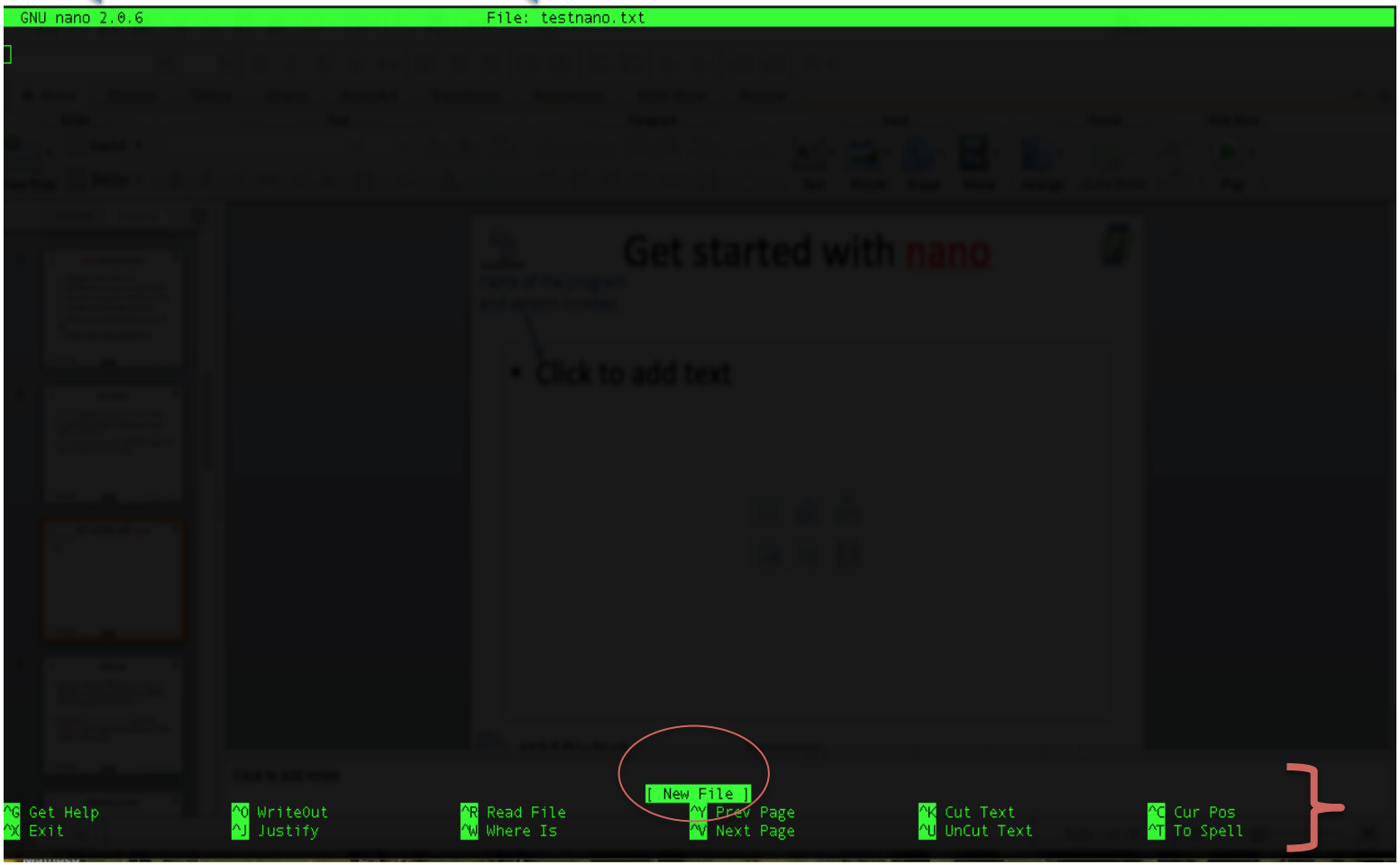
Text editors

- nano: a simple and easy-to-use text editor
- Is installed by default in Ubuntu and many other Linux distributions
- It's a WYSIWYG editor: "what you see is what you get." What you type directly goes into the text input
- vim, emacs, gedit, Geany: excellent programs but do require some learning

Get started with nano

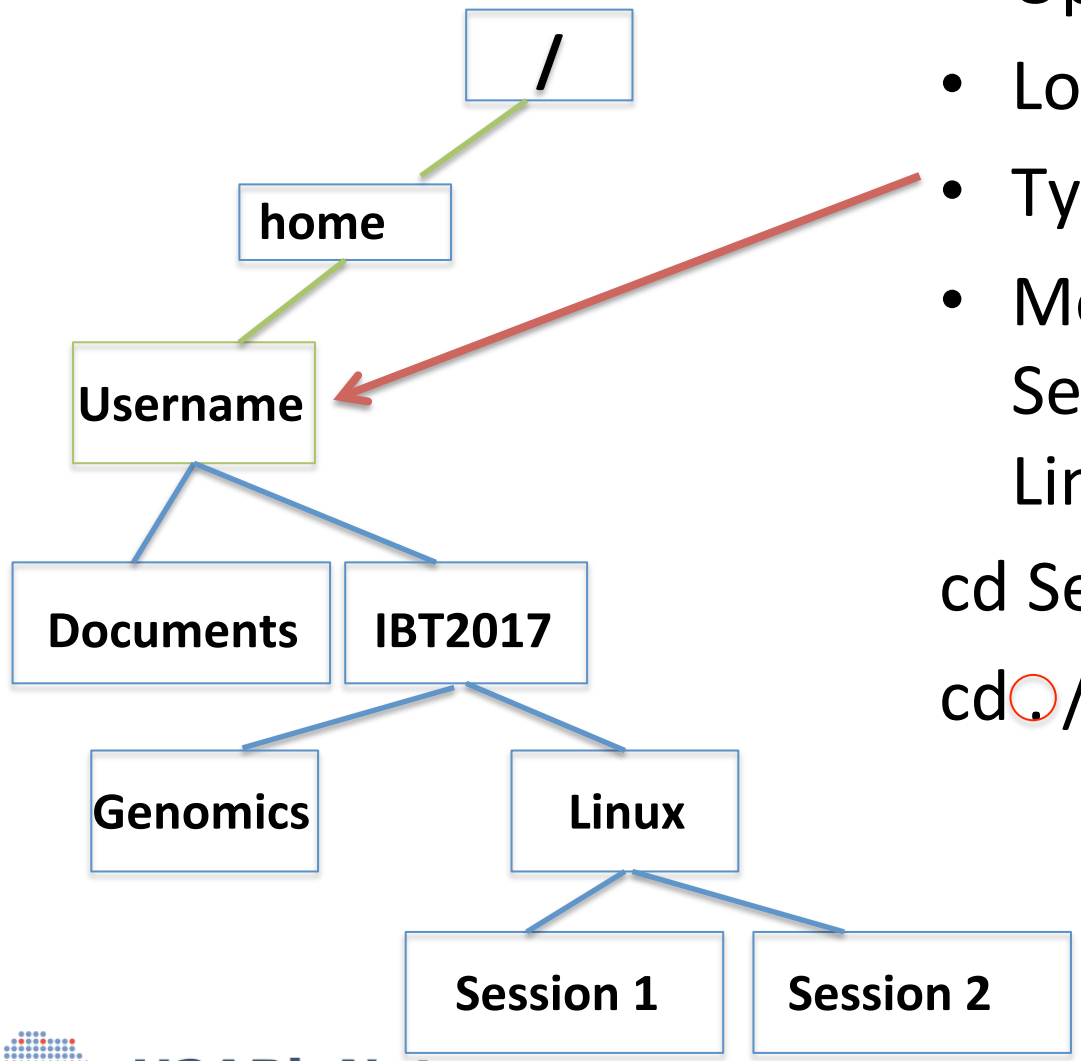
name of the program and version number,

the name of the file you are editing



shortcuts

Let's play with nano



- Open the terminal
- Log in (/home/Username)
- Type pwd to check
- Move to the directory Session1 directory under the Linux

cd Session1?

cd ./IBT2017/Linux/Session1

Get started with nano

- nano file1
- Type “my first test file with webminal”
- Hit enter to move to another line and type “the second line of test”
- One you finish typing, hit Ctrl+x
- Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
- Hit Y
- nano file2
- Type “my second test file with webminal”and any other 4 lines of text

Some nano shortcuts

- To search for a text string, hit Ctrl+W, and enter your search term
- This search can then be cancelled mid-execution by hitting Ctrl+C without destroying your buffer
- Ctrl+X: finish typing and close an open file

Remember: nano pathname

- Opens the file if it's existing already, you can modify and save changes
- Creates a new file in the specified path if it does not exist

H3ABioNet

معهد باستور تونس
Institut Pasteur de Tunis

Part 2

Basic manupulating file commands

Displaying whole content of a file or parts of it *(default + options)*

- **cat**: view the content of a short file

  syntax cat <filename>
- **more**: view the content of a long file and navigate through it

  syntax more <filename>
- **less**: view the content of a long file, by portions

  syntax less <filename>
- **head**: view the first lines of a long file

  syntax head <filename>
- **tail**: view the last lines of a long file

  syntax tail <filename>

View file content: less command

- less command displays a text file content, one page at a time
- Structure: less filename
- Move a page down: either use the page down key or space
- To exit less, type q
- To go to the end of the text file, type g

Head and tail commands

- **head** command displays a text file content, by default: 10 first lines at a time
- Syntax: head <options> <filename>

- **tail** command displays a text file content, by default: 10 last lines at a time
- Syntax: tail <options> <filename>

Basic manupulating file commands

- Copy, move and remove

- cp: copy files and directories

  Structure cp <pathfrom> <path to>

- mv: move or rename files and directories

  Structure mv <pathfrom> <path to>

- rm: remove files and directories

  Structure rm pathname

Copying command: cp

- Simplest form: cp file1 file2

  ➔ Copy the contents of file1 into file2. If file2 does not exist, it is created. Otherwise, file2 is silently overwritten with the contents of file1.

- cp filename dirpath

  ➔ Make a copy of the file (or directory) into the specified destination directory

Other examples: cp

- Add the interactive mode with the option -i
- cp -i file1 file2

  ➔Same as the previous one. However, if file2 exists, the user is notified before overwriting file2 with the content of file1
- cp –R pathdir1 pathdir2

  ➔ Copy the contents of the directory dir1. If directory dir2 does not exist, it is created. Otherwise, it creates a directory named dir1 within directory dir2

Copying command: mv

The mv command moves or renames files and directories depending on how it is used

- **To rename a file:**

mv filename1 filename2

If file2 exists, its contents are silently replaced with the contents of file1. To avoid overwriting, use the interactive mode:

mv -i filename1 filename2

- **To move a file (or a directory) to another directory**:

mv file dirpath

- **To move different files (or a directory) to another directory**:

mv file1 file2 file3 dirpath

- **To move directory to another directory**:

mv dir1 dir2

If dir2 does not exist, then dir1 is renamed dir2. If dir2 exists, the directory dir1 is moved within directory dir2

The rm command

The rm command deletes files and directories

To remove a file:

rm filename

To remove many files:

rm filename1 filename2

Add the interactive mode to prompt user before deleting with –i

rm -i filename1 filename2

Delete directories with all their contents

rm -r dir1 dir2

Be careful with rm !

- Linux does not have an undelete command
- Once you delete something with rm, it's gone!
- You can inflict terrific damage on your system with rm if you are not careful, particularly with wildcards
- Try this trick before using rm: construct your command using ls instead first

Wildcards

- Since the shell uses filenames so much, it provides special characters to help rapidly specifying groups of filenames

- A group of special characters are called wildcards allow selecting filenames based on pattern of characters

Wildcards

Wildcard	Meaning
*	Matches any characters
?	Matches any single character
[!characters]	Matches any character that is not a member of the set characters
[characters]	Matches any character that is a member of the set *characters*. The set of characters may also be expressed as a *POSIX character class* such as one of the following: **[:alnum:]** Alphanumeric characters **[:alpha:]** Alphabetic characters **[:digit:]** Numerals **[:upper:]** Uppercase alphabetic characters **[:lower:]** Lowercase alphabetic characters

Source: http://linuxcommand.org

Wildcards examples

Wildcard	Meaning
a*	Any file name starting with a
*	All possible filenames
A*.fasta	All filenames that begin with A and end with .fasta
????.vcf	Any filenames that contain exactly 4 characters and end with .vcf
[abc]*	Any filename that begins with "a" or "b" or "c" followed by any other characters
[[:upper:]]*	Any filename that begins with an uppercase letter. This is an example of a character class

Download files from the web

- **wget** stands for "web get". It is a command line utility which downloads files over a netwrok
- It supports HTTP, HTTPS, and FTP protocols

Syntax: wget [–options] [URL]

Let's try it:

- Move to the directory Genomics and get the fasta file of P. falciparum from PlasmoDB
- Command: wget http://plasmodb.org/common/downloads/release-9.0/Pfalciparum/fasta/PlasmoDB-9.0_Pfalciparum_BarcodeIsolates.fasta

Thanks

Shaun Aron & Sumir Panji