Part II: reading files, extracting information from files and files permissions

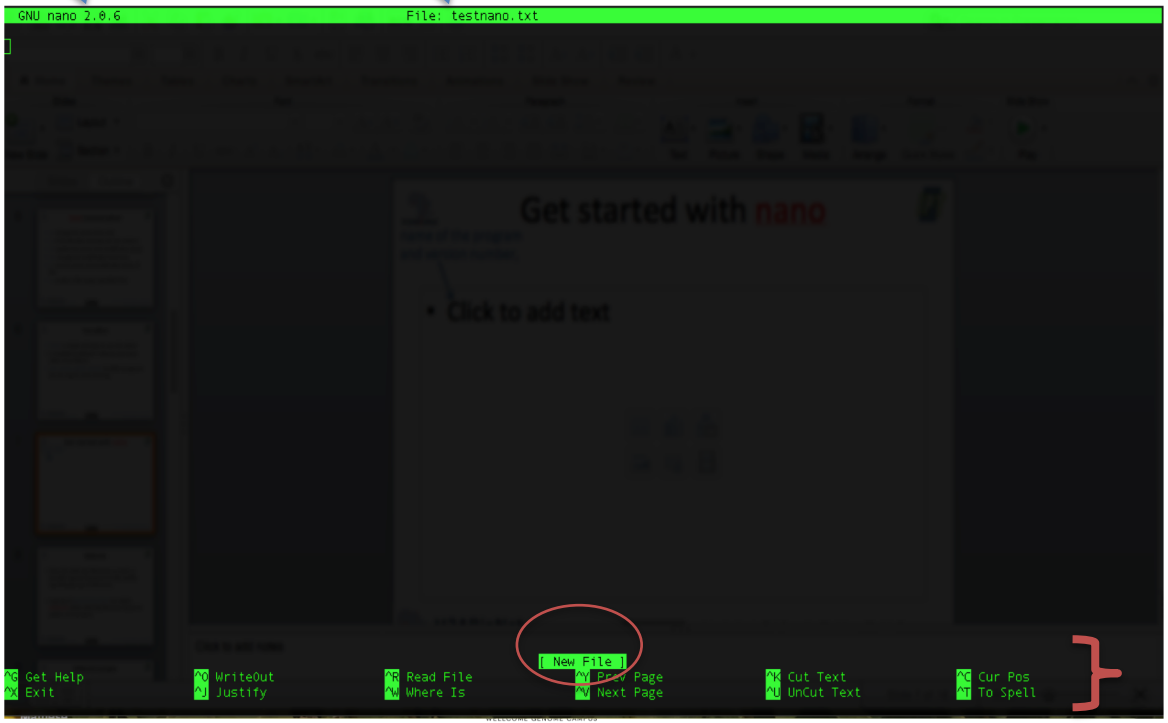
Text editors

- nano: a simple and easy-to-use text editor
- Is installed by default in many other Linux distributions
- gedit is also very easy to use
- vim, emacs, 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

Get started with nano

- nano file1
- Type “my first test file with nano”
- 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

Basic manupulating file commands

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

- **cat**: view the content of a short file
  cat <filename>
- **more**: view the content of a long file and navigate through it
  more <filename>
- **less**: view the content of a long file, by portions
  less <filename>
- **head**: view the first lines of a long file
  head <filename>
- **tail**: view the last lines of a long file
  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

  head <options> <filename>

- **tail** command displays a text file content, by default: 10 last lines at a time

  tail <options> <filename>

use –n to change the number of lines you want to display

Basic manupulating file commands

Copy, move and remove

- cp: copy files and directories

  cp <pathfrom> <path to>
- mv: move or rename files and directories

  mv <pathfrom> <path to>
- rm: remove files and directories

  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

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

Some statistics about your file content: wc command

- wc prints newline, word, and byte counts for each file
  wc <options> <filename>

Some useful options:

- **-c**: print the byte counts
- **-m:** print the character counts
- **-l:** print the newline counts
- For more info about the different commands, remember to use man commandname

Extracting data from files

- **grep**: to search for the occurrence of a specific pattern (regular expression using the wildcards…) in a file

  grep **<pattern>** <filename>

- **cut**: is used to extract specific fields from a file

  cut **<options>** <filename>

grep command

- **grep** ("**g**lobal **r**egular **e**xpression **p**rofile") is used to search for the occurrence of a specific pattern (regular expression…) in a file
- grep outputs the whole line containing that pattern

Example:

Extract lines containing the term sequence from a file:

*grep **sequence** <filename>*

Extract lines that do not contain pattern xxx from a file:

*grep **–v** **sequence** <filename>*

grep example

Let’s create a file named “ghandi.txt” (content below)

cat ghandi.txt

The difference between what we do
and what we are capable of doing
would suffice to solve
most of the world's problems

grep what ghandi.txt

*The difference between **what** we do*
*and **what** we are capable of doing*

grep -v what ghandi.txt

would suffice to solve
most of the world's problems

cut command

- **cut** is used to extract specific fields from a file

  cut <options> <filename>
- For <options> see man
- Important options are
  - **-d** (field delimiter)
  - **-f** (field specifier)

Example:

extract fields 2 and 3 from a file having 'space' as a separator

*cut **-d' ' -f2,3** <filename>*

Practical 2

Instructions

1. Create the file ghandi.txt under Practical2
2. Display lines that contain “what” and those that don’t
3. Count how many words are there in that file
4. Now, we will use 2 files: Pfalciparum.bed and Styphi.fa under practical/Notebooks/files
5. Copy these 2 files under Practical2
6. Check how many lines Pfalciparum.bed contains
7. Display occurrences of the gene PF11_0148 in the bed file
8. Display the names of sequences contained in Styphi.fa (in a fasta file, each sequence is preceded by a description line starting with > and containing the sequence name/ids often followed by additional info)

Additional details about cp and nano

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

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

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