Practical Bioinformatics

Basic Linux

Part 1

Stefan Wyder
stefan.wyder@uzh.ch
URPP Evolution
www.evolution.uzh.ch

Universität Zürich[UZH]

URPP

Why Linux?

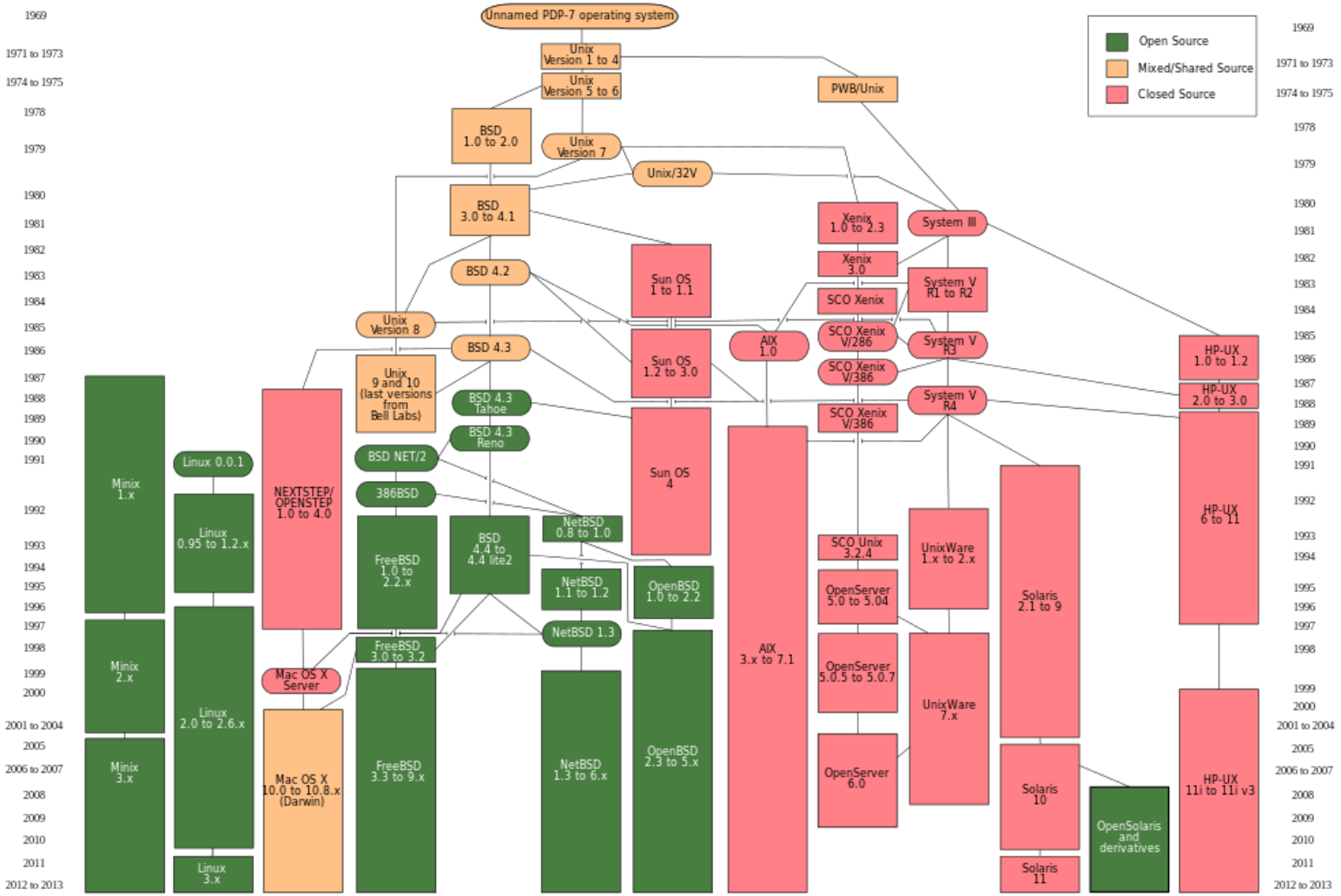
In Bioinformatics

- Many bioinformatics and genomics tools are command-line only and are optimized to run on Linux.
- Very powerful with text files, even large (GBs)

In general

- Experienced Linux users can do more with less effort and much faster
- Multi-user
- Lots of control and customization possibilities
- 0$
- ....

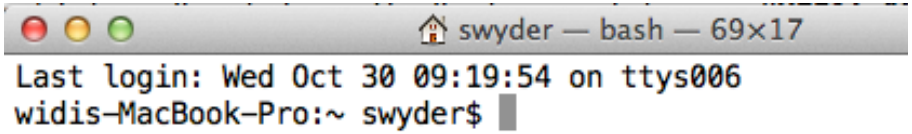
History



http://en.wikipedia.org/wiki/Unix

Why command line?

- to run bioinformatics software
- you can automate tasks more easily
- to interact with high-performance (high-memory) servers
- offers handy tools to work with text files



- The shell is an interactive interpreter: it reads commands, finds the corresponding programs, runs them, and displays output
- Today we use the **Bash shell** (default shell of most Linux distributions and Mac OS X)

The prompt

Prompt

```
eiger-uzh:~ swyder$
```

Hostname

User

Normal user

```
eiger-uzh:~ swyder$
```

Current path
(abbreviated)

Commands

- The tools philosophy was to have small programs to accomplish a particular task instead of trying to develop large monolithic programs to do a large number of tasks.

- "Designed to operate together"
  To accomplish more complex tasks, tools can simply be connected together.

- The shell comprises hundreds of commands, but if you know 25 you can achieve many things

- Commands are abbreviations to type less (ls:list, cp:copy, mv:move)

- Common structure:
  Command -Option(s) Parameter(s)
  ls -l /home/swyder/tmp

Setting options

Command -Option(s) Parameter(s)

ls -l

ls -l -r -h

ls -l -r -h

ls -rlh

ls --reverse --human-readable long options !only GNU tools!

The order of options does not matter
unless they override each other (e.g. sorting)

Options, grep as an example

- $ grep apple fruitlist.txt
  apple
  pineapple

- $ grep -w apple fruitlist.txt (or grep -x)
  apple

- $ grep apple *.txt
  fruitlist.txt:apple
  fruitlist.txt:pineapple
  recipeFruitSalad.txt:1 pineapple
  recipePinaColada.txt:2oz fresh pineapple juice

- $ grep apple fruitlist.txt recipeFruitSalad.txt
  fruitlist.txt:apple
  fruitlist.txt:pineapple
  recipeFruitSalad.txt:1 pineapple

$ grep -v apple fruitlist.txt
banana
pear
peach

Also options to color, to show context, search with compl patterns

Getting help

man <command>

```
man cp

CP(1)            BSD General Commands Manual            CP(1)

NAME
    cp -- copy files

SYNOPSIS
    cp [-R [-H | -L | -P]] [-fi | -n] [-apvX] source_file target_file
    cp [-R [-H | -L | -P]] [-fi | -n] [-apvX] source_file ... target_directory

DESCRIPTION
    In the first synopsis form, the cp utility copies the contents of the source_file to the targe
    contents of each named source_file is copied to the destination target_directory.  The na
    If cp detects an attempt to copy a file to itself, the copy will fail.

    The following options are available:

    -a    Same as -pPR options. Preserves structure and attributes of files but not directory st

    -f    If the destination file cannot be opened, remove it and create a new file, without pro
          permissions.  (The -f option overrides any previous -n option.)
...
```

space: scroll down a page
b: scroll up
q: quit man

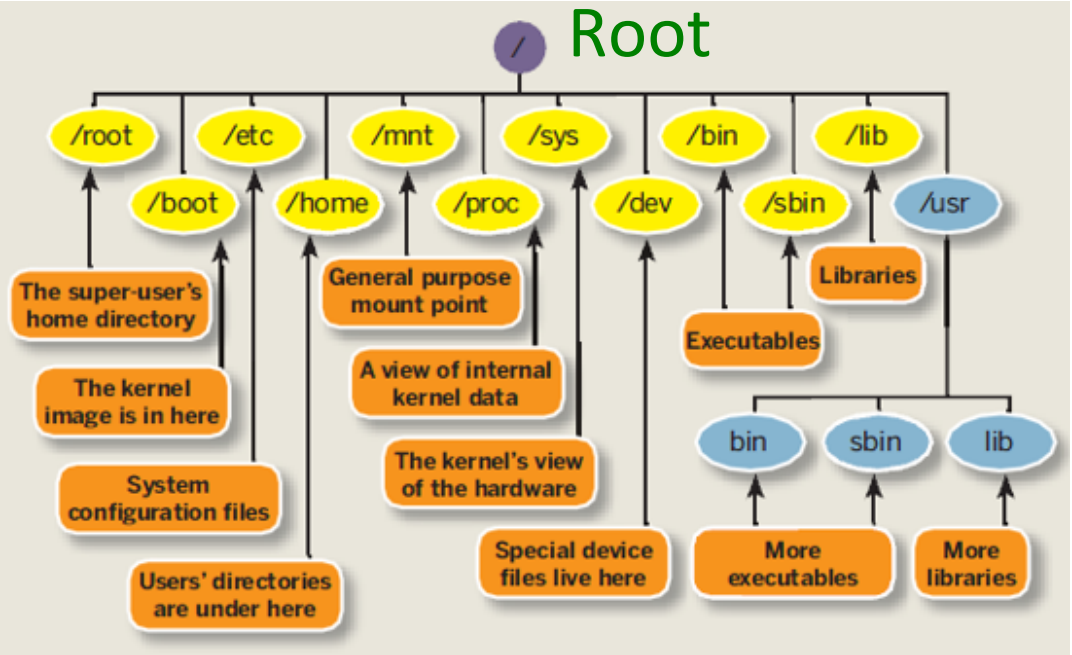
<command> --help
cp --help

a less verbose help
(not working on Mac OS)

The web
http://linuxconfig.org/linux-commands

Working with files and directories

Directory structure



http://www.tuxradar.com/

- Everything the system uses is located somewhere under root '/'.
- Every user has his home directory, e.g. /home/swyder
- Do your work in your home directory
- The system folders can only be modified by the administrator

File and Directory names

- Upper- and lower-case matter
- up to 256 characters long
- Every character except / is allowed. But by convention special characters like $äéÜ are not used.
- Don't use white spaces. Use underscores (_), hashes (-) and dots (.) to separate words
  Oct2013_RNAseq
  ~~Oct2013\ RNAseq~~

Main commands

Command	Meaning
ls	Content of current directory
cd *dir_name*	Change to directory
cd	Change to home directory
cd ~	Change to home directory
mkdir	Make a directory
cp	Copy a file/directory
rm	Delete a file/directory

Working with text files

Main text commands

UNIX has an extensive toolkit for text extraction, reporting and manipulation

Task	Commands
Show	**less**, more, head, tail, cat
Search/Extract	**grep**, **cut**, awk, uniq
Manipulate	**sort**, tr, sed, join, paste
Replace	**tr**, sed
Count	**wc, uniq -c**
Compare	comm, diff

Piping

Piping

Philosophy

"filters“: simple programs which only do 1 thing

the output of a filter is the input of the next

```
grep "mRNA" test.gff | less
```

```
grep -w "gene" test.gff | cut -f 1 | sort | uniq -c
```

Redirection

> Writing the output to a file

>> Appending the output to a file

```
ls > output.txt
grep -w "gene" test.gff | cut -f 1 | sort | uniq -c > output.txt
ls >> output.txt
```

< Reading from file

```
wc < hello.txt > hello_counts.txt
```

Differences in the Shell

Linux - Mac

Differences Linux - Mac

- Mac line breaks are '\r\n' (and Windows: '\r') instead of the standard Linux '\n'
  When working in the command line make sure the files have the right format
  perl -pi -e 's/\r\n?/\n/g' <filename>
  which you could alias and put in the .bashrc

- Mac has no GNU tools by default
  less options – less powerful
  for installation install homebrew
  then: brew install coreutils
  All GNU commands are then installed with the prefix 'g': gls, gcp, gsed,...

- Mac has non-standard folder structure
  e.g. home (~) is /Users/swyder

What next?

Part 2 of the tutorial

- Running scripts
- Permissions
- Installing software (rpm, compiling)
- File/Dir Compression&Extraction
- Automatizing tasks (scripting)
- Connecting with Unix/Linux servers
- (Search Patterns, Regular Expressions)
- Backup (rsync)

Sources & Links

Acknowledgements

- Some exercises are from von Mering group (IMLS, UZH)

Material

- SIB course http://edu.isb-sib.ch/course/view.php?id=41
- O'reilly Books http://oreilly.com/linux/
- Video tutorials (~100 min) http://software-carpentry.org/v4/shell/index.html
- Cheatsheet http://www.embnet.org/sites/default/files/quickguides/guideUNIX.pdf

Intro

- Unix has been developed in the late 60‘s
- Linux (specifically:the Linux kernel) was created in 1991 by Linus Torvalds. Together with the GNU tools and libraries, this established a freely available Unix on regular PC hardware

- For the sake of simplicity I use Linux even when Unix would be more appropriate from now on
- Over time, graphical user interfaces have been added making Linux easier to use.

Many Linux Distributions

Ubuntu, Debian, CentOS, RedHat, Fedora, Slackware, SuSE, Darwin
Main differences are the graphical system (KDE vs Gnome, support durance