Tools for scientific computing 1

Holger Schulz (IPPP)
holger.schulz@durham.ac.uk

February 26, 2016

Course overview — tentative list

1. Introduction to `bash`
2. Introduction to `python`
3. Data presentation with `python/matplotlib`
4. Typesetting with $\LaTeX$

1. Version control (subversion, mercurial, git)
2. Working remotely (ssh, scp, tramp mode)

This document:

`https://users.hepforge.org/~holsch/Teaching/Computing/201516/L01.pdf`

Operating system (OS)

- In this course, `Debian` is used as OS
- `Debian` is a “Distribution” of `Linux`
- http://www.debian.org
- Other distributions: fedora, ubuntu, gentoo, ...
- `Linux` is
  - free (GNU General Public License)
  - the most widely-used OS in physics (e.g. all the computing at CERN)
  - quite handy for an academic career inside/outside physics

First steps

Log on to machine

- Machines in `PHY216` are dual boot
- Select `Linux desktop` in the boot menu (use arrow keys)
- Use `CIS` user name and password when prompted

First steps

Log on to machine

- Machines in `PHY216` are dual boot
- Select `Linux desktop` in the boot menu (use arrow keys)
- Use `CIS` user name and password when prompted

Opening a shell (synonym for terminal)

- Navigate with your mouse to `Applications` (top left corner)
- Select `System tools`
- Click `MATE terminal`

For some reason, the default shell on machines in `PHY216` is `csh`. Although most of the commands and syntax are similar, `bash` is far more wide-spread among `Linux` distributions than `csh`. To get a `bash` shell, type `bash` and hit `[Enter]` whenever you open a new terminal.

Remote log in

Log in from an external terminal:

- `ssh CISUSER@mira.dur.ac.uk` — terminal only
- `ssh -X CISUSER@mira.dur.ac.uk` — with graphic forwarding ("X")

Moving around

- By default, your current directory will be your `HOME` directory
- To list the contents of your current directory, type `ls`
- Get the full path by typing `pwd`
- If your are unsure about your user name, type `whoami`
- Some useful information about the system you are running on is obtained with the command `uname`
  - Most commands have options, to see them type `ACOMMAND --help`
  - E.g. `uname -a` or `hostname`
  - Extensive help is usually available via man pages: `man ACOMMAND`
- To change your current to another directory, use `cd`:
  - `cd Public` moves you to the directory "Public" (convince yourself by using `pwd`)
  - To go back one directory, relative to the current directory, use `cd ..`
  - To go back to the previous directory use `cd -`
  - To go back to your HOME directory use `cd`

Creating, copying, moving, deleting

Files

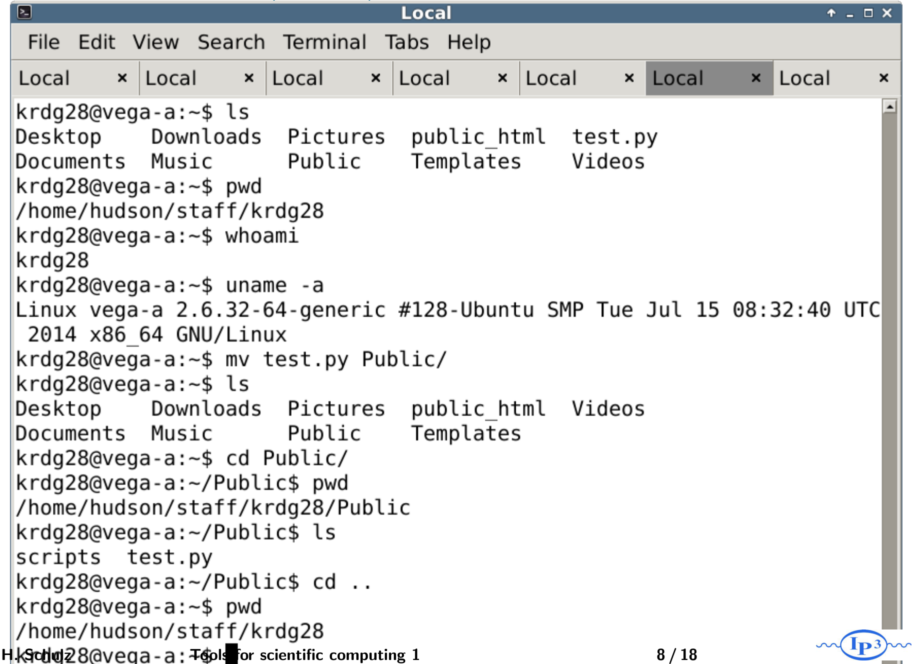
- Creating empty files is done using `touch`, e.g.:
  - `touch testfile.txt`
- To move/rename files use `mv SOURCE DESTINATION`:
  - `mv testfile.txt Public` to move (Public is a directory)
  - `mv testfile.txt renamedfile.txt` to rename
- Copying files is just as easy using `cp SOURCE DESTINATION`
- Deleting is performed via `rm FILE` — BE CAREFUL, there is no waste-basket in linux

Creating, copying, moving, deleting

Directories

- Creating directories is done using `mkdir`, e.g.:
  - `mkdir work` for single directories
  - `mkdir -p work/Physics/QED/Exercises` for trees
- It is good practice to have a few directories in your home directory with meaningful names
- To move directories use `mv SOURCE DESTINATION`, e.g.:
  - `mv work/Physics/QED/Exercises .`
- Copying directories is just as easy using `cp -r SOURCE DESTINATION`
- Deleting is performed via `rm -r DIRECTORY` — BE CAREFUL, there is no waste-basket in linux

Hello world (bash)


Local
File Edit View Search Terminal Tabs Help
Local × Local × Local × Local × Local × Local × Local ×

```
krdg28@vega-a:~$ ls
Desktop    Downloads  Pictures  public_html  test.py
Documents  Music      Public    Templates    Videos
krdg28@vega-a:~$ pwd
/home/hudson/staff/krdg28
krdg28@vega-a:~$ whoami
krdg28
krdg28@vega-a:~$ uname -a
Linux vega-a 2.6.32-64-generic #128-Ubuntu SMP Tue Jul 15 08:32:40 UTC
 2014 x86_64 GNU/Linux
krdg28@vega-a:~$ mv test.py Public/
krdg28@vega-a:~$ ls
Desktop    Downloads  Pictures  public_html  Videos
Documents  Music      Public    Templates
krdg28@vega-a:~$ cd Public/
krdg28@vega-a:~/Public$ pwd
/home/hudson/staff/krdg28/Public
krdg28@vega-a:~/Public$ ls
scripts  test.py
krdg28@vega-a:~/Public$ cd ..
krdg28@vega-a:~$ pwd
/home/hudson/staff/krdg28
krdg28@vega-a:~$
```

Copy and paste

- Copy and paste is a little different in Linux

- To copy something, use the mouse:
  1. point to the beginning of a text
  2. click and hold the left mouse button
  3. move the mouse to highlight (release left mouse button when done)
  4. whatever is highlighted goes into a virtual clipboard

- To paste anywhere:
  1. Left click into the window where you want to paste to
  2. press the middle mouse button to paste

For strings without spaces, such as URLs, a double-click will also highlight and copy to clipboard

Downloading files and extracting archives

- If you know the URL to a file, you can download it to your current directory using `wget`
- E.g.:

```
wget https://users.hepforge.org/~holsch/Teaching/Computing/L_01/l_01.tar.gz
```

- This file is a compressed archive, a so-called “tarball”
- To unzip it, use `tar`: `tar xzf l_01.tar.gz`

Downloading files and extracting archives

- If you know the URL to a file, you can download it to your current directory using `wget`
- E.g.:

```
wget https://users.hepforge.org/~holsch/Teaching/Computing/L_01/l_01.tar.gz
```

- This file is a compressed archive, a so-called “tarball”
- To unzip it, use `tar`: `tar xzf l_01.tar.gz`

Tab completion

- Bash offers the powerful feature of tab completion
- Instead of typing full paths and file names, just type a few letters and hit `[TAB]` to automatically complete e.g. file names
- Hit `[TAB]` repeatedly to see options, e.g. in your HOME directory, try tab completion when only entering “P”
- You can save a lot of time and typing

Viewing files

- There is quite a number of options to view contents of files:
  - $\boxed{\texttt{less FILENAME}}$ opens a lightweight viewer, exit by typing “q”
    This is useful to browse though text files

Viewing files

- There is quite a number of options to view contents of files:
  - `less FILENAME` opens a lightweight viewer, exit by typing "q"
    This is useful to browse though text files
  - `cat FILENAME` dumps all lines of a file to STDOUT, i.e. the shell
    This is useful when you know that this is a short file and when redirecting STDOUT

Viewing files

- There is quite a number of options to view contents of files:
  - $\boxed{\texttt{less FILENAME}}$ opens a lightweight viewer, exit by typing “q”
    This is useful to browse though text files
  - $\boxed{\texttt{cat FILENAME}}$ dumps all lines of a file to STDOUT, i.e. the shell
    This is useful when you know that this is a short file and when redirecting STDOUT
  - $\boxed{\texttt{head FILENAME}}$ dumps the first few lines of a file to STDOUT
    This is useful for log-files and when redirecting STDOUT

Viewing files

- There is quite a number of options to view contents of files:
    - `less FILENAME` opens a lightweight viewer, exit by typing "q"
      This is useful to browse though text files
    - `cat FILENAME` dumps all lines of a file to STDOUT, i.e. the shell
      This is useful when you know that this is a short file and when redirecting STDOUT
    - `head FILENAME` dumps the first few lines of a file to STDOUT
      This is useful for log-files and when redirecting STDOUT
    - `tail FILENAME` dumps the last few lines of a file to STDOUT
      This is useful for log-files and when redirecting STDOUT

Redirecting STDOUT

- To redirect STDOUT to a file, use the operator `>`
  e.g. `cat text_01.txt > redirectedoutput.txt`
- A lot of linux tools support reading in the output from another tool[1]
- The operator used is `|` ("pipe").
- The structure can be arbitrarily complex:
  `CMD1 | CMD2 | CMD3 | CMD4 | CMD5` and so forth
  e.g. `wc` is useful to count lines and words:
  `cat text_01.txt | wc -l` prints the number of lines in text_01.txt

[1] this is kind of the linux philosphy, one small programme for one small task

Manipulating STDOUT

- If you want very specific information from a file, use `grep`:
  `grep Roxanne text_01.txt` for single words or “strings”
  `grep "red light" text_01.txt` for more complicated strings
- Task: produce a command that counts all occurrences of “Roxanne” in text_01.txt

Manipulating STDOUT

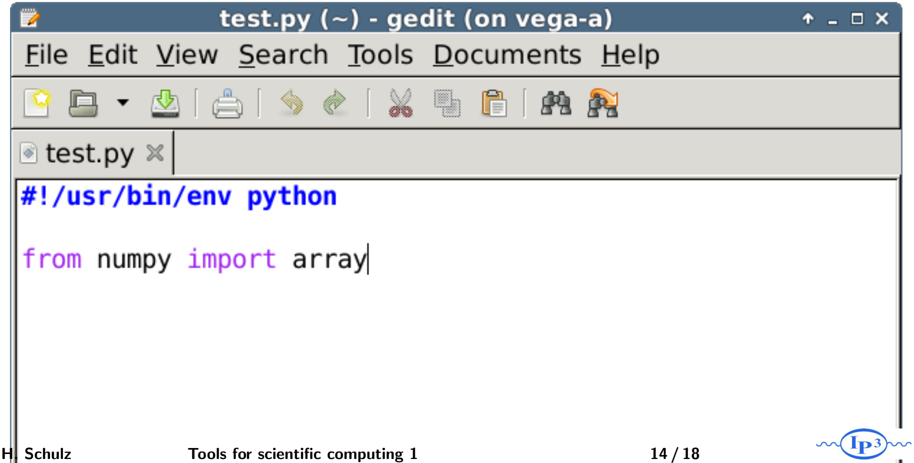
- If you want very specific information from a file, use `grep`:
  `grep Roxanne text_01.txt` for single words or "strings"
  `grep "red light" text_01.txt` for more complicated strings
- Task: produce a command that counts all occurrences of "Roxanne" in text_01.txt

Search and replace

- Search and replace is done in linux using `sed`
- The structure is `sed "s|STRING|REPLACEMENT|g" FILENAME`
  or when piping to sed:
  `cat FILENAME | sed "s|STRING|REPLACEMENT|g"`
- E.g. `sed "s|Roxanne|Cameron|g" text_01.txt | sed "s|the red light|some more weight|g"`

Editors

- The most important software to work with
- Open/create files, edit and save changes
- Important: syntax highlighting (when editing code)
- Popular choices: `gedit` for beginners, `vim` and `emacs` for advanced users


test.py (~) - gedit (on vega-a)
File Edit View Search Tools Documents Help
test.py

```
#!/usr/bin/env python

from numpy import array
```

Bash scripts

- Bash commands can be put into files, called “scripts”
- A file in linux can be made executable using `chmod a+x FILENAME`
- The script is then executed by calling `./FILENAME`
- This is extremely powerful when having to deal with tasks that are performed often

Bash scripts

- Bash commands can be put into files, called “scripts”
- A file in linux can be made executable using `chmod a+x FILENAME`
- The script is then executed by calling `./FILENAME`
- This is extremely powerful when having to deal with tasks that are performed often

Example script

```
#!/bin/bash
echo "You are" ${USER}
echo "Running program:" ${0}
echo "With command line arguments:" ${1}
echo "and" ${2}
echo ${1} | sed "s|hello|goodbye|g"
```

Remember, make it executable, then run

```
./script.sh "hello world" "this is the second argument"
```

Formative work

- Write a bash script `countWords.sh` that
  - takes two arguments:
    1. `a string`
    2. `a text file`
  - counts all occurrences of the `string` in the `text file`, regardless of the case of `string`
- such that $\boxed{\text{./countWords.sh thunder text\_02.txt}}$ prints `20`

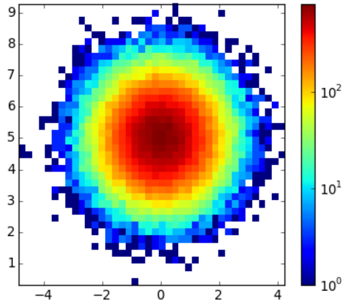
More bash

- Almost every problem you will encounter has been reported, discussed and solved by someone else
- Google is an excellent tool to help you solve your bash problems — use it!
- General hint: prefer google search results mentioning `stackoverflow`

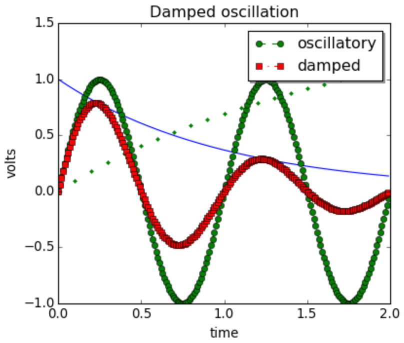
- So-called "cheat sheets" are available for a large number of linux programs, they give an excellent overview of commands, e.g. google for "bash cheat sheet"

Next week

- A little bit of data-mangling with python and plotting
- Send a mail with requests



−4
−2
0
2
4
1
2
3
4
5
6
7
8
9
10^0
10^1
10^2


Damped oscillation
oscillatory
damped
volts
time