

Computational Physics

Lecture 2

sieversj@ukzn.ac.za

Brief Unix Recap

- How would you find out what directory you're in?
- What is the difference between “cd /home” and “cd home”?
- What would “cd ..” do?
- What would “ls *.jpg” do?
- How can I find out more about Unix commands?
- What is the difference between “more” and “cat”?
- How can I look at the last 10 lines of a file?
- How can I delete a file? A directory?

Let's Make a Useful File

- Whenever you start a new shell, a resource file is used. For us, that will be “\${HOME}/.bashrc” (bourne-again shell resources).
- Let's make it so we can get through the proxy. Let's set “http_proxy” so we can get out from the command line.
- “export http_proxy=http://<username>:<password>@bcproxy.ukzn.ac.za:8080”.
- NEVER write your password in plain text. Proxy forcing us now, though.
- You can put this in your .bashrc. How would you check if new terminals register this?
- ftp_proxy and https_proxy should also be set to the same value.
- If this worked, “wget www.cnn.com” should go grab the CNN homepage

File Permissions

- You can specify who can look at your files
- Each file has 3 settings - Read, Write, and eXecute
- There are 3 classes of people - user, group, world
- “ls -l” shows permissions.
- to change, use “chmod” +/- turns permissions on/off, ugo sets permissions on user/group/others, and rwx sets read/write/execute.

```
Jonathans-MacBook-Pro:lecture2 sievers$ echo hello class > foo
Jonathans-MacBook-Pro:lecture2 sievers$ ls -l foo
-rw-r--r--+ 1 sievers  staff  12  3 Apr 09:15 foo
Jonathans-MacBook-Pro:lecture2 sievers$ chmod o-r foo
Jonathans-MacBook-Pro:lecture2 sievers$ ls -l foo
-rw-r-----+ 1 sievers  staff  12  3 Apr 09:15 foo
Jonathans-MacBook-Pro:lecture2 sievers$ cat foo
hello class
Jonathans-MacBook-Pro:lecture2 sievers$ chmod u-r foo
Jonathans-MacBook-Pro:lecture2 sievers$ ls -l foo
--w-r-----+ 1 sievers  staff  12  3 Apr 09:15 foo
Jonathans-MacBook-Pro:lecture2 sievers$ cat foo
cat: foo: Permission denied
Jonathans-MacBook-Pro:lecture2 sievers$
```

Make any file that has a password in it not readable to others!

Version Control

- Let's role play.

Version Control ctd.

- What did we just learn? It's important to be able to undo changes
- Merging changes from different places/different people important. Should be done in a systematic way. If I change one part of a file while you change another, would like to combine those changes automatically.
- What happens if someone steals your computer? The more places code lives, the safer you are.

Version Control 3

- Several version control (VC) systems exist, 3 most common are CVS, SVN, and git. We will use git.
- VC lets many people edit files. A set of files being tracked is called a “repository.” A VC system can keep your local files in a repository synced with the central repository.
- When you are done making changes, you “commit” them to your local repository, then “push” them to the central repository. When someone else makes changes, you “pull” them from the central repository.
- If you both edited the same part of a file, the VCS will throw an error and tell you to fix the conflict by hand.
- VCSs usually require your local repository to be up-to-date with the central one before you’re allowed to push changes.

Git

- git was developed by Linux Torvalds (the “Li” in linux)
- very powerful/flexible VCS.
- Basic commands: “git init” will initialize an empty repository. “git add <files>” will add those files to the repository.
- When ready, “git commit -a -m “<message>” “ will commit your changes to your repository.
- “git push” can push to a central repository.
- “git history” will tell you the messages you saved in your commits.
- “git revert” will let you go to a previous version of the repository.

Github

- Github provides free repository support. Much of the code released today is done on github.
- You can make an account, then github will walk you through the steps you need to do what you want.
- “git clone” will copy a repository. I have made a github page for this class - you can get the slides plus a linux tutorial cheat sheet there.
- In the browser, go to “<https://github.com/ukzncompphys>”

Click on lecture 1

GitHub

Search or type a command

ExploreFeaturesEnterpriseBlog

Sign upSign in

Contributions

Repositories

Public Activity

Follow

ukzncompphys

Joined on Mar 31, 2014

0followers

0starred

0following

Popular repositories

welcome

This contains a welcome message for UKZN computational physics students.

0

★

lecture1

Slides from Lecure 1

0

★

Public contributions

Apr

May

Jun

Jul

Aug

Sep

Oct

Nov

Dec

Jan

Feb

Mar

M

W

F

Summary of Pull Requests, issues opened and commits.

Learn more.

Less

More

0 Total

Mar 31 2013 - Mar 31 2014

Year of Contributions

0 days

Rock - Hard Place

Longest Streak

0 days

Rock - Hard Place

Current Streak

Contribution Activity

Period: 1 Week

Slides from Lecure 1

 1 commit





 1 branch


 0 releases


 1 contributor


 branch: master ▾


lecture1 / 


initial commit		
 sievers authored 7 minutes ago	latest commit 8a8fb868a6 	
 compphys_1.key	initial commit	7 minutes ago
 linux_boot_camp.pdf	initial commit	7 minutes ago


 Code

 Issues 0


 Pull Requests 0

 Pulse


 Graphs


 Network

HTTPS clone URL

https://github.com/ 

You can clone with [HTTPS](#) or [Subversion](#). ?

 Clone in Desktop

 Download ZIP

PUBLIC



ukzncompphys / lecture1



Star

0



Fork

0

Slides from Lecure 1



1 commit



1 branch



0 releases



1 contributor



branch: master ▾

lecture1



initial commit



sievers authored 7 minutes ago

latest commit 8a8fb868a6



compphys_1.key

initial commit

7 minutes ago



linux_boot_camp.pdf

initial commit

7 minutes ago

<> Code



Issues

0



Pull Requests

0



Pulse



Graphs



Network

HTTPS clone URL

https://github.com/



You can clone with [HTTPS](#) or [Subversion](#). ?



Clone in Desktop



Download ZIP

Then Clone the Repository

- “git clone <link from github>” will get you the slides.
- <https://github.com/ukzncompphys/lecture1.git> is the link to Tuesday’s slides in case you can’t get to it.
- <https://github.com/ukzncompphys/lecture2.git> is the link to today’s slides.
- If this doesn’t work, your proxy settings haven’t been set correctly.

You might want to stay at same computer during course. However, if you haven’t, just can just “git pull/clone” from a different machine and be set up in seconds.

Quick Introduction to Python

(Don't worry, we'll do this again next week)

How did I learn?

<https://docs.python.org/2.7/tutorial/index.html>

Python Variables

```
#intro to some built-in python types
#(PS - everything following a # is a comment and ignored)
a='hello and welcome to python' #assign a string
b="is this a string?"           #single or double OK
print "a is " + a + ' and b is ' + b #have a look!

c=5 #assign an integer to c
d=3.14159 #assign a float to d
e=[c,d,a] #e is a list - can hold all sort of things
e[0]='3' #python is zero indexed

#f is a tuple - cannot be changed
f=(3,4,5)
#f[0]=4 #this is an error
g=(1==2)
#look at your types
print 'type of g is ' + repr(type(g))
-uuu:---F1 python_variables.py Top L16 (Python)-----
Wrote /Users/sievers/talks/python/python_variables.py
```

```
MACBOOKPRO-BA0F:python sievers$ python
Python 2.7.1 (r271:86832, Jun 16 2011, 16:59:05)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> execfile('python_variables.py')
a is hello and welcome to python and b is is this a string?
type of g is <type 'bool'>
>>> ^D
MACBOOKPRO-BA0F:python sievers$
```

Dictionaries, methods, and our first 'for'

```
#lets look at dictionaries, then some controls
d=dict() #this initialized an empty dictionary
d['key1']=5
d['pi']=3.14159265
d[5]='keys can be numbers, too'
#Python associates methods with objects, so
#they can be "self-aware." Methods are accessed
#using the period. We can use this to learn about
#our dictionary
print d.keys() #you can have a look at the keys in a dictionary
#try also typing "d." and then hitting tab to see what
#else you can do with the dictionary
```

```
#now lets loop over keys
for key in d.keys():
    print repr(key) + ' is ' + repr(d[key])
print 'did we need an end?' #python indents instead of 'end' or {}

-uuu:---F1 python_dicts.py All L17 (Python)-----on darwin.
```

```
>>> execfile('python_dicts.py')
['key1', 'pi', 5]
'key1' is 5
'pi' is 3.14159265
5 is 'keys can be numbers, too'
did we need an end?
>>> type(d)
<type 'dict'>
>>>
```



```

#We can bring in external modules to do more things
import numpy #For NUMerical PYthon
#numpy is the standard for math-intensive python

tot=0
for x in range(0,10): #range is built-in, puts a sequence into a list
    tot=tot+x
a=numpy.arange(0,10); #numpy has a version that returns a numpy array
tot2=a.sum() #many useful methods are attached to numpy arrays
print "My sums are " + repr(tot) + " and " + repr(tot2)
b=a*2 #we can vectorize calls - this is *much* faster than a loop
print b
print b[1:3] #we can slice a numpy array
print b[-2:] #negative indices start from the end.
import numpy as np #we can call modules whatever we want
from numpy import sin #we can also import functions from modules
c=sin(b);d=np.sin(b); #we can put multiple commands on a line, split with ;
print np.sum(np.abs(c-d))
-uuu:---F1 numpy_intro.py All L9 (Python)-----

```

```

>>> execfile('numpy_intro.py')
My sums are 45 and 45
[ 0  2  4  6  8 10 12 14 16 18]
[2 4]
[16 18]
0.0
>>>

```

Let's meet
modules, import,
numpy, and slicing

Writing a module/class in python

```
import numpy
def get_evens(vec): #we can define functions
    a=vec[numpy.mod(vec,2)==0] #slice along the evens
    #We can do an alternative slice into a list
    b=[x for x in vec if numpy.mod(x,2)==0]
    return (a,b) #send a tuple back
#now let's define a class. We'll fill it with even numbers
class MyClass:
    #__init__ tells python how to start up a class
    def __init__(self,imax=10): #you can pass multiple values to a function
        vec=numpy.arange(imax)
        a,b=get_evens(vec) #get the even numbers
        self.vec=a #these will now be fields in the class
        self.list=b
    def double(self): #we can write functions that operate on our class
        self.vec=2*self.vec #let's double the numpy vector
```

```
>>> import numpy,myfile
>>> a=myfile.MyClass()
>>> print a.vec
[0 2 4 6 8]
>>> print a.list
[0, 2, 4, 6, 8]
>>> a.double()
>>> print a.vec
[ 0  4  8 12 16]
```

Tutorial Problems

- How can you list all files in a directory, with the most recent one displayed last? (5)
- How can you display the first 25 lines of a file? How can you display just lines 16 through 25? (5)
- How can you search for all commands relating to python? Hint - it will be a use of “man”. (5)
- Make a new text file with your name, email address, what you’re working on for your honours project, plus a few things you’d like to learn in the course. Initialize a git repository and commit this file. (5)
- Put your answers to the other tutorial questions into another text file. Add this to the repository also. (5)
- Make a github account and push the repository onto github. Email me your github name so I can have a look at your file/answers. (10)