Only administer as prescribed abuse outside of seminar may lead to UNIX and Pyhton addiction for which no remedy is known

Meeting schedule Monday May 6 to Friday May 10, 10am-noon, 2-4 in Dirac 152.

This course has no prerequisite but may need new thinking and follow instructions, the form of the course is unclear and not well defined yet.

We will learn about basic UNIX commands and basic Python commands to allow manipulation of files and perhaps are able to construct pipelines to transform data from one format into another

SC/remedial

Scientific Computing DSL 152

You and your friends

Florida State University

UNIX and Python 120 MIN

Common brands: macosx, linux

TAKE 2 CLASSES A DAY FOR A WEEK

Qty: 10 Refills: as needed

Phone: (850) 645-1324
RX # 200532260110
Prescriber: PETER BEERLI

Workshop about Python and basic UNIX within a short week

Week May 6-10 ➟ Monday ➟ 10-12 UNIX 2-4 ➟ UNIX ➟ Tuesday ➟ 10-12 Introduction to Python; what is it and why should you use it ➟ 2-4 ‘Hello world’ and strings/words/texts ➟ Wednesday ➟ 10-12 using modules and writing more complex program ➟ 2-4 File input/output ➟ Thursday ➟ 10-12 Stringing things together ➟ 2-4 Plotting data ➟ Friday ➟ 10-12 Exploration of modules ➟ 2-4 Outlook, Review.

Today

- Overview of UNIX
- First contact
- Details of most important commands

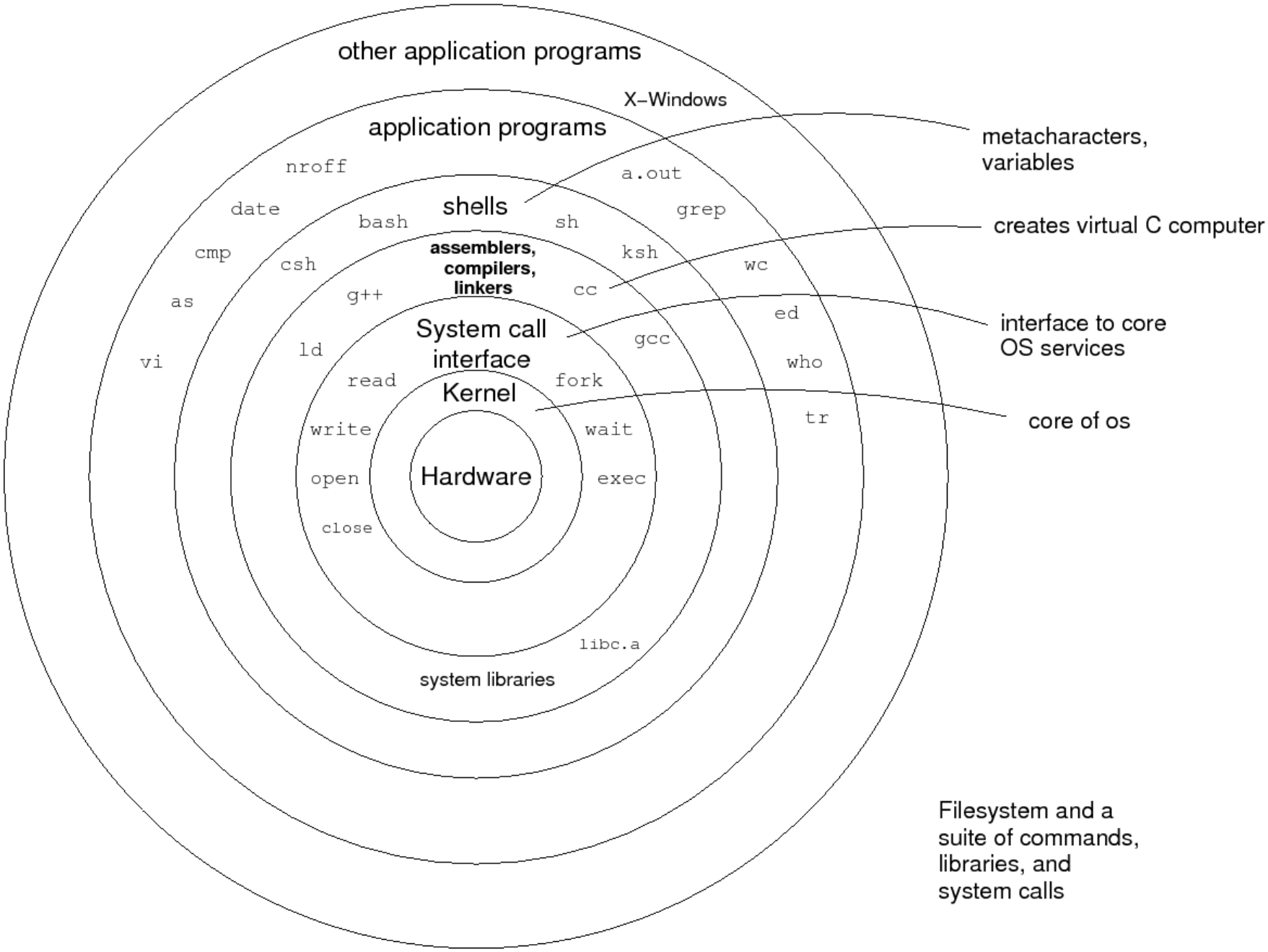
- Editing in the UNIX commandline
- Practice
- Pipelines
- Extracting the stuff you want
- Synthesis

Unix

Operating system
Multitasking
Multi-user
1969
AT&T
Running on PDP-7

http://www.faqs.org/docs/artu/ch02s01.html

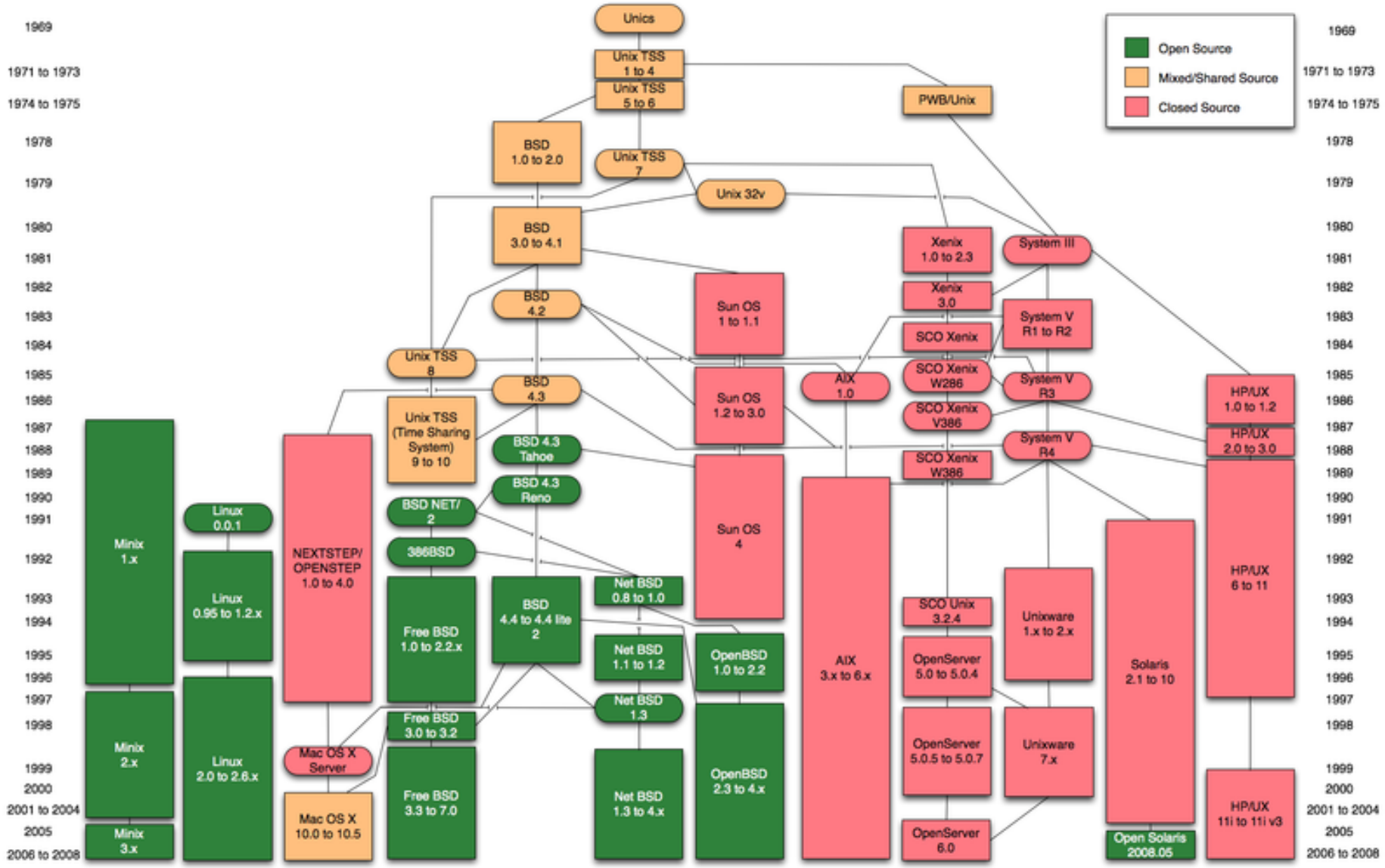
AIX
IBM
ORACLE
SOLARIS
HP-ux11i v3



Conceptual Architecture of UNIX SYSTEMS

http://academic.udayton.edu/SaverioPerugini/courses/cps346/lecture_notes/UNIXphilo.html

Evolution of Unix and Unix-like OS



From [Wikipedia]

Linux

Unix-like free Operating system
1991 ~
Linus Tovalds & many others
Linux distribution

- Unix-like operating system based on "*Linux kernel*"

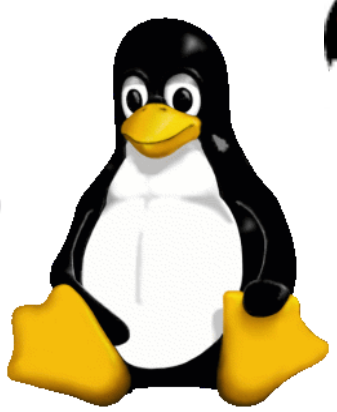
Used by all of world top 10 Supercomputers(June, 2011)

ubuntu

debian

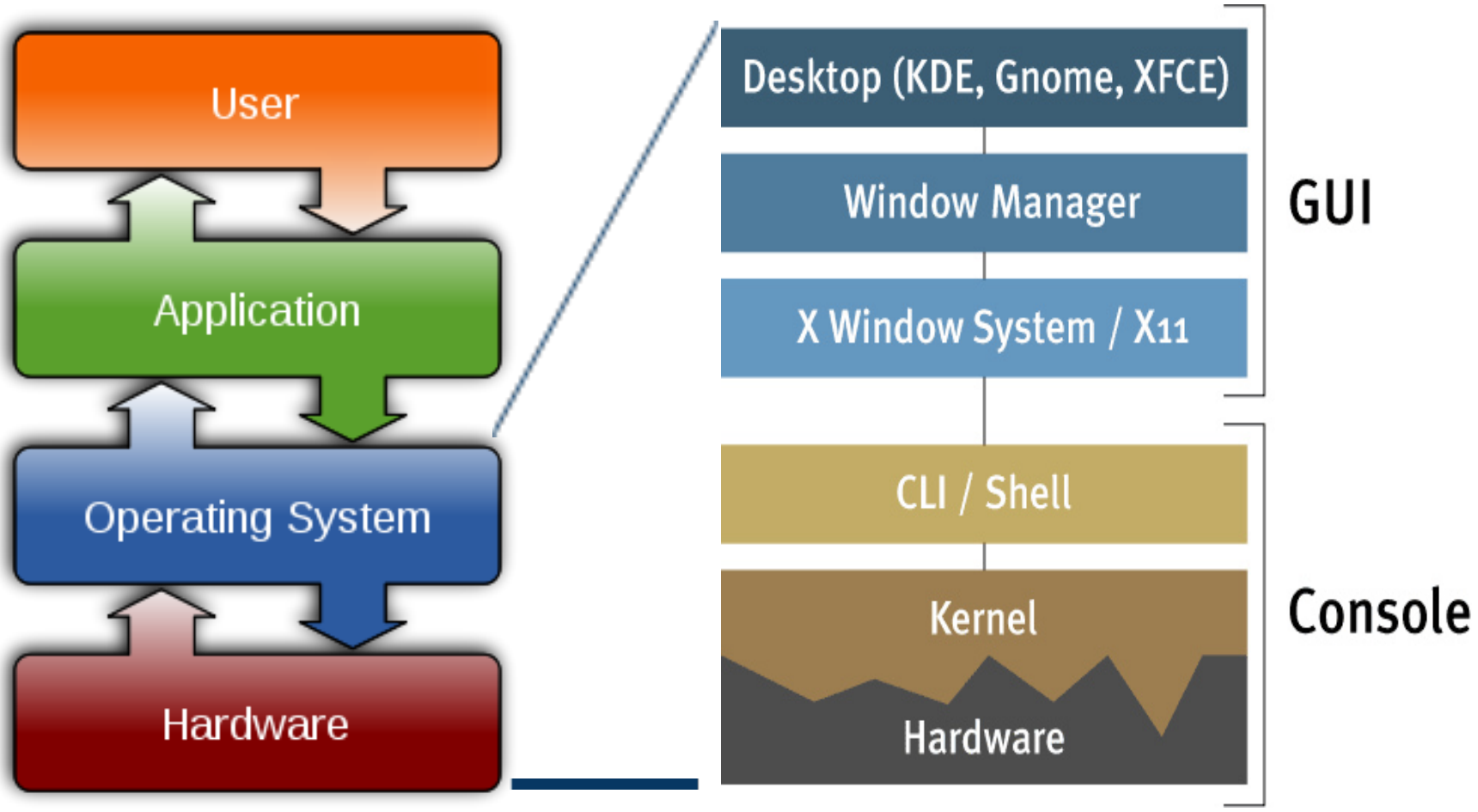
redhat.

CentOS



openSUSE

Operating System



Remote Access

	Windows	Linux/Mac
Login	putty	ssh
Copy files	winscp	scp
Check availability	ping	ping

Free download from

putty
http://www.chiark.greenend.org.uk/~sgtatham/putty/
*Configuration http://sc.tamu.edu/help/access/windows.php

winscp
http://winscp.net/

A UNIX command line consists of the name of a UNIX command (actually the "command" is the name of a built-in shell command, a system utility or an application program) followed by its "arguments" (options and the target filenames and/or expressions). The general syntax for a UNIX command is

```
$ command -options targets
```

Here `command` can be though of as a verb, `options` as an adverb and `targets` as the direct objects of the verb. In the case that the user wishes to specify several options, these need not always be listed separately (the options can sometimes be listed altogether after a single dash).

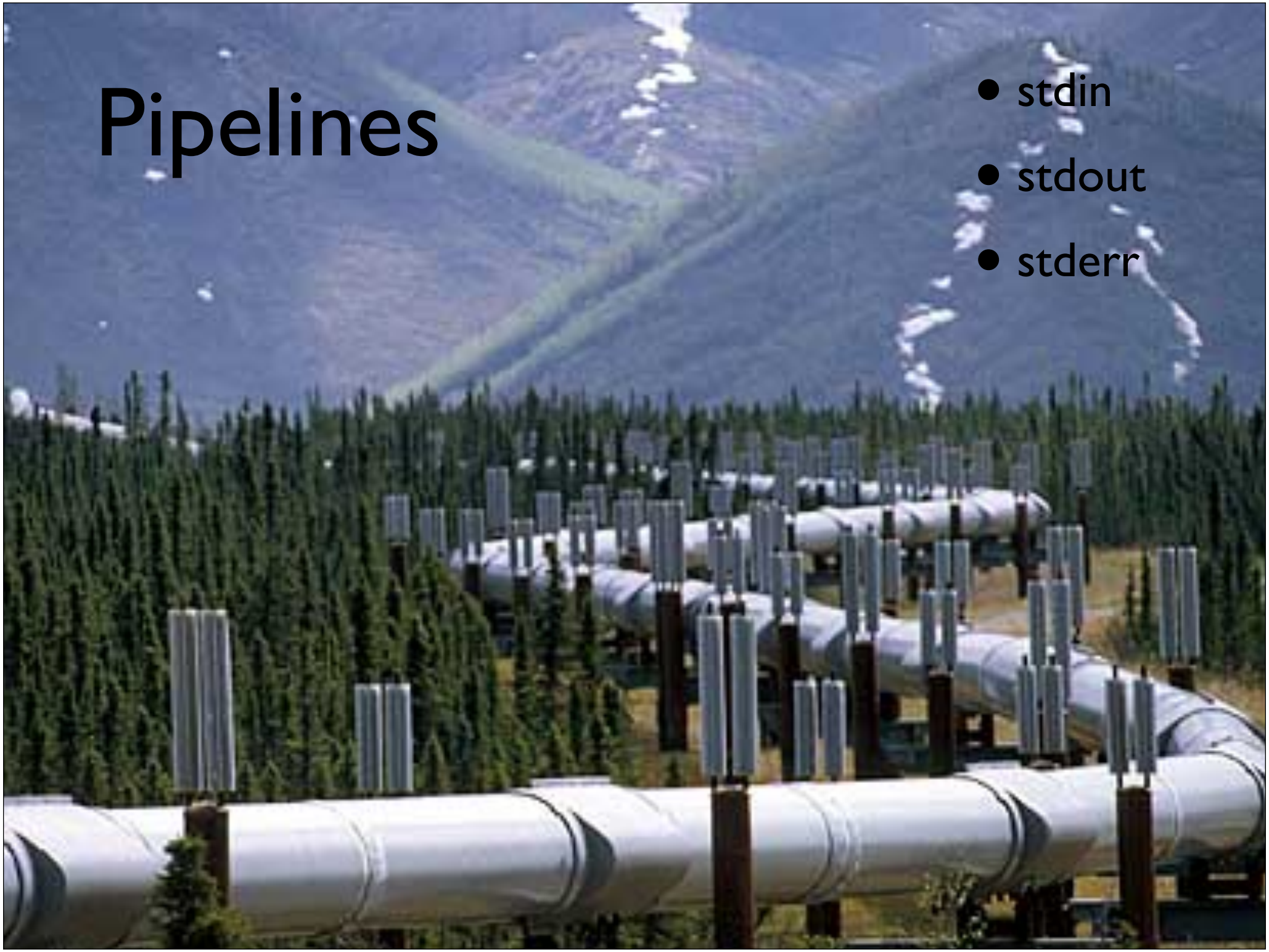
1. Log on a Linux machine or connect to one from a Windows machine (e.g. click on the Exceed icon and then use putty to connect to the server `kiwi`. Enter your login (user name) and password at relevant prompts.
2. Enter these commands at the UNIX prompt, and try to interpret the output. Ask questions and don't be afraid to experiment (as a normal user you cannot do much harm):

   - `echo hello world`
   - `passwd`
   - `date`
   - `hostname`
   - `arch`
   - `uname -a`
   - `dmesg | more`
     (you may need to press `q` to quit)
   - `uptime`
   - `who am i`
   - `who`
   - `id`
   - `last`
   - `finger`
   - `w`
   - `top` (you may need to press `q` to quit)
   - `echo $SHELL`
   - `echo {con,pre}{sent,fer}{s,ed}`
   - `man "automatic door"`
   - `man ls` (you may need to press `q` to quit)
   - `man who` (you may need to press `q` to quit)
   - `who can tell me why i got divorced`
   - `lost`
   - `clear`
   - `cal 2000`
   - `cal 9 1752` (notice anything unusual?)
   - `bc -l` (type `quit` or press Ctrl-d to quit)
   - `echo 5+4 | bc -l`
   - `yes please`
     (you may need to press Ctrl-c to quit)
   - `time sleep 5`
   - `history`

UNIX editors

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

Most common basic UNIX editors


Pipelines
• stdin
• stdout
• stderr

```
cat hello.txt | sort | uniq
```

```
cat hello.txt | grep "dog" | grep -v "cat"
```

To redirect standard output to a file instead of the screen, we use the > operator:

```
$ echo hello
hello
$ echo hello > output
$ cat output
hello
```

In this case, the contents of the file `output` will be destroyed if the file already exists. If instead we want to append the output of the echo command to the file, we can use the >> operator:

```
$ echo bye >> output
$ cat output
hello
bye
```

To capture standard error, prefix the > operator with a 2 (in UNIX the file numbers 0, 1 and 2 are assigned to standard input, standard output and standard error respectively), e.g.:

```
$ cat nonexistent 2>errors
$ cat errors
cat: nonexistent: No such file or directory
$
```

You can redirect standard error and standard output to two different files:

```
$ find . -print 1>errors 2>files
```

or to the same file:

```
$ find . -print 1>output 2>output
```

or

```
$ find . -print >& output
```

Standard input can also be redirected using the < operator, so that input is read from a file instead of the keyboard:

```
$ cat < output
hello
bye
```

You can combine input redirection with output redirection, but be careful not to use the same filename in both places. For example:

```
$ cat < output > output
```

will destroy the contents of the file `output`. This is because the first thing the shell does when it sees the > operator is to create an empty file ready for the output.

Grep