Unix: commands, scripts, useful features
Steps and Stages

- Unix: commands, aliases, scripts, emacs
- Programming: languages
- Algorithms
- Running codes and handling data
- Analysis
- Astrophysics: interpretation
Useful Commands

ls          : list directory contents
ls -l        : long list of files:
Useful Commands

`ls` : list directory contents
`ls -lht` : long list in ‘human’ format sorted by time.
## Useful Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls</code></td>
<td>list directory contents</td>
</tr>
<tr>
<td>`ls -lht</td>
<td>head -15`</td>
</tr>
</tbody>
</table>

```plaintext
antigona.~> ls -lht | head -15
```
```
total 21M
-dwrxrwxr-x 2 aklypin aklypin 4.0K Jan 13 16:40 TMP
-drw-r--r-- 1 aklypin aklypin 440K Dec 22 19:13 BDM.tar
-drw-r--r-- 1 aklypin aklypin  413 Oct  2 11:59 id_rsa.pub
-drwtxrwxr-x 4 aklypin aklypin  4.0K Sep 11 17:41 MAIL
-drwxrwxrwx 14 aklypin aklypin  4.0K Aug 23 18:27 HOLTZMAN
-lwxrwxrwx  1 aklypin aklypin  4 Aug 23 13:31 mail -> MAIL
-drwxrwxrwx 25 aklypin aklypin  4.0K Mar 11 2012 HTML
-drwxr-xr-x 25 aklypin aklypin 1000 4.0K Nov 21 2011 CODE
-drwxr-xr-x  1 aklypin aklypin  78 Nov 21 2011 stgmsrun.log
-drwxr-xr-x  1 aklypin aklypin 263K Nov 21 2011 gms16402.ps
-drwxr-xr-x  1 aklypin aklypin  269 Nov 21 2011 stgms.log
-drwxrwxr-x  4 aklypin aklypin  4.0K Nov 21 2011 NASA
-drwxrwxr-x  3 aklypin aklypin  4.0K Oct 31 2011 NAMES
-drwxr-xr-x  1 aklypin aklypin  31K Jul  6 2011 gms26122.pdf
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls -alht</code></td>
<td>long list of files including hidden files, which start with <code>.</code></td>
</tr>
</tbody>
</table>
Useful Commands

ls -l *.f90 | wc : count number of files with extension .f90 (all Fortran-90 files)

whoami - print effective userid

cd - : go back to previously visited directory
Useful Commands

**top** : what is going on. Jobs, memory, usage of cpu

```
top - 19:55:11 up 193 days, 6:24, 1 user, load average: 4.00, 4.00, 4.00
Tasks: 309 total, 5 running, 304 sleeping, 0 stopped, 0 zombie
Cpu(s): 25.0%us, 0.0%sy, 0.0%nI, 75.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 16437302k total, 699068k used, 9446708k free, 342840k buffers
Swap: 32784556k total, 54320k used, 32710236k free, 6217164k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
24897 redmonds 25 0 18812 15m 1204 R 99.9 0.1 344:58.40 gcm2.0_s_1.0
24898 redmonds 25 0 18812 15m 1204 R 99.9 0.1 344:43.43 gcm2.0_1.0
24902 redmonds 25 0 18812 15m 1200 R 99.9 0.1 344:23.81 gcm2.0_no_gw_1.
24906 redmonds 25 0 18812 15m 1200 R 99.9 0.1 343:45.61 gcm2.0_no_gw_s_
26059 aklypin 15 0 17132 1404 912 R 0.3 0.0 0:00.08 top
25862 aklypin 15 0 92272 1960 1172 S 0.0 0.0 0:08.41 init
  0 root 15 0 10364 636 544 S 0.0 0.0 0:08.31 migration/0
  2 root 15 0 0 0 0 S 0.0 0.0 0:00.05 ksoftirqd/0
  3 root 15 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/0
  4 root 15 0 0 0 0 S 0.0 0.0 0:00.55 migration/1
  5 root 15 0 0 0 0 S 0.0 0.0 0:00.22 ksoftirqd/1
  6 root 15 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/1
  7 root 15 0 0 0 0 S 0.0 0.0 0:01.39 migration/2
```

**df -h** : what is going on with disks

```
antigona.$> df -h
Filesystem Size Used Avail Use% Mounted on
/dev/sdd3 19G 5.0G 13G 28% /
/dev/sdd5 240G 52G 176G 23% /antigona
/dev/sdd1 122M 26M 91M 22% /boot
tmpfs 7.9G 0 7.9G 0% /dev/shm
/dev/md0 2.7T 2.2T 583G 79% /cosmology3
/dev/md1 2.7T 1.6T 1.2T 58% /antigona2
/orpheus:/astro/local 2.1T 555G 1.5T 27% /home/local
/orpheus:/astro/users 2.1T 555G 1.5T 27% /home/users
astronomy:/eurelica/mail 1.3T 400G 847G 33% /home/mail
```

**who** : lists users currently using computer

```
erebos.$> who
root pts/1 2013-01-01 23:34 (:1.0)
root pts/2 2013-01-01 23:34 (:1.0)
root pts/5 2013-01-02 00:21 (alma244.11n.gw)
got pts/12 2013-01-04 12:15 (leo.aip.de)
got pts/13 2013-01-04 13:28 (leo.aip.de)
nil pts/17 2013-01-07 10:15 (draco.aip.de)
```
directory /proc has information about your system
Useful Commands: files

wc file.name : count lines in file file.name

file CatAccSm...dat has 76 lines, 243 characters and 3116 bites

head -20 file.name : show first 20 lines of file.name

grep Galaxy Catalog.DAT | wc : count how many lines in file Catalog.DAT have text ‘Galaxy’

use awk for more elaborate search
Useful Commands

sort file.name : sort file file.name, print sorted file

sort -g --key=2 -r file.name : sort lines in the file file.name by values in the second column in reverse order (lines with the largest values come first) using ‘normal’ values

grep 100.25 Catalog.dat | sort -g -r --key=12 > sorted.file.dat

awk : use awk for complicated search of text in a file.
In this example awk searches file out.list.dat for lines where the ninth column is in the range 983-996 and the tens column in the range 582-594. Every line, which satisfied the condition is printed. The output is piped to file cluster3.dat. One can print only some columns (say, print $1, $2, $10). Line numbers also can be printed as well as some simple combination

awk '{if($9> 983 && $9 <996 && $10>582 && $10<594 ) print}' out.list.txt > cluster3.dat
Useful Commands

ln : make links between files

This is a handy command which allows you to use files as if they are in your directory, but in reality they are just pointers to other files. For almost all practical points, linked files are real files. You see them listed in ‘ls’. When you read or write to them, you really read and write to the files. However, the actual files are not in your directory. They are somewhere else.

ln -s /home/antigona/aklypin/Catalog*.DAT .

Directory /home/antigona/aklypin/ has many files, which names start with ‘Catalog’ and end with ‘DAT’. Instead of copying all the files, ‘ln -s’ makes pointers to the files. Note that writing to the files will change content of real files. However, removing the links, with ‘rm Catalog*.DAT’ will remove only the links, but not real files.
Useful Commands

- **history**: list of previously typed commands

```bash
praesepe./home/hyades/aklypin/NBODY>history | head
23   cd TMP
24   du -hs
25   rm -f *
26   exit
27   ls /home/hyades
28   df -h
29   cd /home/hyades/aklypin/
30   ls
31   mkdir bigMD
```

- **history > grep mkdir**: will make list of all typed commands, that had text ‘mkdir

- **history > tail -10**: will make list of 10 last typed commands.
When you login, you use cpu and graphics of your desktop/laptop, but your home directory is on /home/users, which is mounted on departmental server -- a different computer.

This is an example. My computer is antigona. The prompt tells me that I am using antigona. `pwd` shows that I am in my home directory and use system disk /home/users. I do not use cpu of that server, which has the disk, but I go over the net to bring information from that computer.

Now make it more complicated. My desktop is antigona. I login to praesepe and cd to /home/hyades/aklypin/NBODY. What resources am I using?

```plaintext
Graphics: antigona
CPU : praesepe
Disk : hyades
```
Useful Environment

setting system prompt. If you use bash, set `PS1= ....`

- `34m` codes the color of the prompt. 34 - blue, 30 - black, 38 - red
- `1;34m` - bold blue, `0:34m` - normal blue
- `\h` : print host name
- `\w` : print directory name
- `\t` : print time
- `\u` : print userid
Useful Environment

$HOME : has path to home directory

export WORK=/hyades/aklypin/Catalogs : Variable WORK now has path to my work directory

cd $WORK : change directory to the value assigned to WORK

ln -s $WORK/*.exe . : makes soft link to all my executable files in WORK
Making life more comfortable

aliases are a great way to get organized and to make convenient short-cuts for system-independent commands

Examples:

making ‘human’ commands:

```bash
alias dir=`ls -l`
alias print=`lpr`
alias copy=`cp`

almost mandatory:

alias cp=`cp -i`
alias mv=`mv -i`
alias rm=`rm -i`

short-cuts:

alias m=`more`
alias x=`emacs`
alias e=`emacs -nw`

connections:

alias leo1=`ssh -X leo1.aip.de`

navigation:

alias CODE=`cd $HOME/CODE`
alias RUN1=`cd $scratch/RUN1`
Add lines with ‘alias’ statements to your shell profile.

For bash it is .bashrc:

```bash
export OMP_NUM_THREADS=4
PS1="[\033[1;31m\]h.\w>\[\033[0m\] "
alias m='more'
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias dir='ls -l'
alias aphrodite='/usr/bin/ssh aphrodite.nmsu.edu -l yourUID'
alias praesepe='/usr/bin/ssh -Y praesepe.nmsu.edu -l yourUID'
alias CODE='cd $HOME/CODE'
alias NBODY='cd $HOME/CODE/NBODY'
```
Connecting to other computer(s):

Use `ssh` and `scp`
To avoid typing you password again and again, use `ssh-keygen` and then `ssh-add`. In this case you type your pass only once.
Steps:
1) `ssh-keygen -t rsa`
2) take ...pub file and `scp` it to another computer
3) add it to a file `.ssh/authorized-keys`
4) run `ssh-agent bash` on your computer
5) run `ssh-add`. You will be prompted to give passphrase.
Next time you login to your computer make (4) and (5). Your system may be setup so that you do not need (4).

Connection with the external computer may be slow. To edit files use `emacs -nw`, which makes editing possible with emacs even with very slow connection. You lose some functionality of emacs: cannot use `buttons`. Yet, it is better than using vi.
Emacs is a standard editor, which you can find in any unix or Mac OS system. It has many options and can be adjusted to make your editing more efficient.

The best way to customize emacs is to make changes to .emacs
Edit file .emacs and add those lines there
Shell Scripts

• Simulations often create many files. For example, code may produce many snapshots.

• Analysis of many snapshots requires special tools of handling repetitive tasks

• Archiving and retrieving of many models and snapshots
Simple example:

```bash
#!/bin/tcsh -f
set file1 = ( 6010 6020 6030 6040 6050 6070 8560 )
set file2 = ( 7560 7570 7580 7590 )
set file3 = ( 9999 9990 9980 9970 )

foreach file ( $file1 $file2 $file3)
    $HOME/Code/PM/PMgalaxy3 << EOF
    $file
    EOF
end

exit
```

This script creates a list of snapshots (variables file1, 2,3). Then it calls executable PMgalaxy3 many times and feeds it with current snapshot number. It works fine given the number of the snapshots is not large.

Place the text in a file ‘run.bat’. Make it executable (chmod a+x run.bat). Then just run it as if it is a unix command:

run.bat
Programming with Bash: loops

```bash
#!/bin/bash
for i in $(ls); do
  echo item: $i
done
```

this gives list of files in current directory with every entry starting with 'item:'

```
praesepe/home/hyades/aklypin/Box200/CATALOGS~/bin/example.bat
item: Part.0003.DAT
item: Part.0014.DAT
item: plot14.ps
item: plot3.ps
item: points.sm
```

```
#!/bin/bash
files=$(ls *DAT)
for i in ${files} ;
do
  echo $i
done
```

this gives a list of files with extension DAT in current directory
Passing arguments to script

$0 : name of the script
$1 and $2 are two parameters passed to the script

The script copies files from one set of name to the other depending on the value of the the second parameter.
The End