Introduction to UNIX

Where there is a shell, there is a way.

Jeroen Engelberts jeroen.engelberts@surfsara.nl
Consultant Supercomputing
Outline

Introduction to UNIX
• What is UNIX?
• (Short) history of UNIX

Login to Lisa
• With username and password
• With a key pair

Hands on – Let’s Play!
• Install UNIX tool on your laptop
• Create key pair
• Login to Lisa
• First step on a UNIX, or rather Linux, system
About SURFsara

- SURFsara offers an integrated ICT research infrastructure and provides services in the areas of computing, data storage, visualization, networking, cloud and e-Science.
- SARA was founded in 1971 as an Amsterdam computing center by the two Amsterdam universities (UvA and VU) and the current CWI
- Independent as of 1995
- Founded Vancis in 2008 offering ICT services and ICT products to enterprises, universities, and educational and healthcare institutions
- As from 1 January 2013, SARA – from then on SURFsara – forms part of the SURF Foundation
- First supercomputer in The Netherlands in 1984 (Control Data Cyber 205). Hosting the national supercomputer(s) ever since.
What is UNIX?

• Operating System
  • Program that controls all other parts of a computer system
  • Allocates computer’s resources and schedules tasks
  • Allows the user to use the facilities provided by the system
  • Essential to all computer systems
• Multi-User, Multi-Tasking
  • Multiple users have multiple tasks running simultaneously
• Designed to be machine independent
• Setup as a software development environment
History of UNIX (1)

Developed by AT&T Bell Labs, starting 1969
- Ken Thompson (sitting) and Dennis Ritchie – on a PDP-11
In their own words:

My Dropbox:  
https://www.dropbox.com/s/x7p7yblyh8vy4os/Intro_UNIX.mkv

Original from YouTube:  
https://www.youtube.com/watch?v=JoVQTPbD6UY
Remote Login (SSH) to SURFsara

Authentication on (some) SURFsara systems

*Lisa & Cartesius*
- username/password
- key pair

*HPC Cloud*
- key pair (only!)

*Grid*
- username/password (local clusters)
- Grid certificate (for other national and international clusters)

NB Grid certificates (X.509) will not be discussed here
Login to Lisa – with username (1)
Login to Lisa – with username (2)
Login to Lisa – with username (3)

```
Last login: Wed Jan  6 01:16:21 on ttys001
MLT0073:~ jeroene$ ssh jeroene@lisa.surfsara.nl
SURFsara

  Welcome to Lisa

  This is a private computer facility. Access for any reason must be specifically authorized by the owner. Unless you are so authorized, your continued access and any other use may expose you to criminal and/or civil proceedings.

  Information:  http://www.surfsara.nl/systems/lisa/news

jeroene@lisa.surfsara.nl's password:
```
Login to Lisa – with username (4)

The LISA team wishes you all a computational 2016!

Questions and remarks to helpdesk@surfsara.nl or call 020-8001400

Filesystem Quota Used Avail Use% Server
/home/jeroene 200.0 GB 21.53 GB 178.47 GB 10% fs5
Asymmetric Encryption – key pair (1)

Creating Private and Public Keys
Asymmetric Encryption – key pair (2)
Login to Lisa – with key pair (1)

```
Last login: Wed Jan 6 01:43:40 on ttys000
MLT0073:~ jeroene$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/jeroene/.ssh/id_rsa): 
```
Login to Lisa – with key pair (2)

Last login: Wed Jan  6 01:43:40 on ttys000
MLT0073:~ jeroene$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/jeroene/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/jeroene/.ssh/id_rsa.
Your public key has been saved in /Users/jeroene/.ssh/id_rsa.pub.
The key fingerprint is:
The key's randomart image is:

```plaintext
-----[ RSA 2048]-----
  . 0.0 0 .
  + 0 0
  . 0
  s 0
  +
  .
  .
  e
  +
```

MLT0073:~ jeroene$
Login to Lisa – with key pair (3)

Introduction to UNIX
Login to Lisa – with key pair (4)
Login to Lisa – with key pair (5)

The LISA team wishes you all a computational 2016!

Questions and remarks to helpdesk@surfsara.nl or call 020-8001400

Filesystem Quota Used Avail Use% Server
/home/jeroene 200.0 GB 21.53 GB 178.47 GB 10% fs5
jeroene@login2:~$ vi .ssh/authorized_keys
Login to Lisa – with key pair (6)

Introduction to UNIX
Login to Lisa – with key pair (7)

The LISA team wishes you all a computational 2016!

Questions and remarks to helpdesk@surfsara.nl or call 020-8001400

Filesystem Quota Used Avail Use% Server
/home/jeroene 200.0 GB 21.53 GB 178.47 GB 10% fs5
jeroene@login2:~$ vi .ssh/authorized_keys
jeroene@login2:~$ chmod 644 .ssh/authorized_keys
jeroene@login2:~$
Questions
Install UNIX tool on your laptop

**Windows**
- MobaXterm ([http://mobaxterm.mobatek.net](http://mobaxterm.mobatek.net))

**Mac OS X**
- Terminal (pre-installed in /Applications/Utilities)
- XQuartz ([http://www.xquartz.org](http://www.xquartz.org))

**Linux**
- You are already well equipped!
Hands on – Let’s Play!

Next slides copied from previous years (with thanks to)
- Willem Vermin (retired)
- John Donners (colleague)
After successful login:

```
wiltest@login4:~$
```

- **Login name**: wiltest
- **Name of the system**: login4

The system is ready to process your commands
First command

You type:

date

followed by an 'Enter'

wiltest@login4:~$ date
Mon Mar 15 14:26:17 CET 2010
wiltest@login4:~$
More commands

w who is logged in
whoami who am I?
Whoami
abc an unknown command
uname name of the system
Say goodbye

logout
Behind the screens

Q: What is interpreting the things you type in at the $ prompt?
A: a program called 'shell'

You type:
\texttt{date}

The shell tries to find a program called 'date' and takes care that the system executes it.
An interactive program

You type:

```
bc
4+7
5*9+10
quit
```

The program 'bc' is listening to your keyboard, not the shell.

bc has no prompt.
Flags, parameters

The working of most programs can be influenced by flags (parameters), for example try:

```
date -u
bc -q
```
The shell gets this:

```
date -u
```

Shell locates program 'date', puts '-u' in a place where the program 'date' can find it, and starts the program 'date'. 'date' finds the flag '-u' and changes its internal workings.

Try:

```
date -x
```
Files and folders

We made some examples for you

Type:

svn export https://subtrac.surfsara.nl/userdoc/examples/lisatutorial
ls (list subdirectory)

Type:

ls

wiltest@login4:~$ ls
lisatutorial
wiltest@login4:~$

ls -l

wiltest@login4:~$ ls -l
total 3
-drwx------ 3 wiltest wiltest 3 2010-03-15 15:45 lisatutorial
wiltest@login4:~$

directory = folder
ls and cd (change directory)

Type:

`ls -l lisatutorial`

```
wiltest@login3:~$ ls -l lisatutorial
total 2
drwx------ 2 wiltest wiltest 3 2010-03-22 14:05 c-example
drwx------ 2 wiltest wiltest 3 2010-03-22 14:05 jobs
drwx------ 2 wiltest wiltest 4 2010-03-22 14:05 scripts
drwx------ 2 wiltest wiltest 5 2010-03-22 14:05 simple
```

`cd lisatutorial`

```
wiltest@login4:~$ cd lisatutorial
wiltest@login4:~/lisatutorial$
```

Notice the change in prompt

`~` = home directory

`/` separates directory names
More cd and ls

Type:

Conclusion:

brings you back to your home directory

cd lisatutorial
ls
cd simple
ls -l

wiltest@login4:~/lisatutorial/simple$ ls -l
total 5
-rw------- 1 wiltest wiltest 129 2010-03-15 15:39 bcin
-rw------- 1 wiltest wiltest 221 2010-03-15 15:39 file1
-rw------- 1 wiltest wiltest  94 2010-03-15 15:39 file2.txt

Q: Why 5 and not 3?
A: Not shown . and .. (current and parent directory)
Hidden files

File names, starting with '.', are not shown by default.

Use the '-a' flag of ls to make them visible:

```
ls -a -l
ls -l -a
ls -la
```

Often, you can combine flags

Where am I? **pwd** print working directory

Type:

```
pwd
```
Creating a directory

```
mkdir mydir
ls -l
```

```
  drwx------ 2 wiltest wiltest   2 2010-03-19 13:49 mydir
```

Permission bits!

Each file or directory has 9 permission bits associated:

- nr 0: '-' normal file
- nr 123: read, write, execute for owner
- nr 456: idem for group
- nr 789: idem for others

nr 0: 'd' directory
nr 123: 'which files', 'create files', 'cd to' for owner etc
Permission bits, example, chmod

Type:

cd
cd mydir
cd
chmod -x mydir
ls -ld mydir
cd mydir
chmod +x mydir
cd mydir

Remove x-bit from mydir
-d flag of ls: show properties of mydir, not the contents
Will fail
Content of files

This looks like something you could feed into bc ...

```
Another command: cat

cd

# this is a file for bc.
# note: bc ignores lines starting with #
# let's make a complicated computation:
3+128*9877-123*(45+98)
```

...
Standard input, output and error

Every program (bc, shell, ...) has three predefined input/output files associated:

Standard input (stdin): normally your keyboard
Standard output (stdout): normally your screen
Standard error (stderr): normally your screen

stderr is for error messages (in general)

In the 'simple' directory, type:

```
bc < bcin
```

Instructs the shell, that bc should take stdin from file 'bcin'
More redirection

In the 'simple' directory:

- redirection of stdin to a file
  - `bc < bcin`

- redirection of stdout to a file
  - `bc < bcin > bcout`

- output of cat goes to input of bc
  - `cat bcout`

- 'pipe': output of cat goes to input of bc
  - `cat bcin | bc | cat | cat | cat | cat | bc | bc | cat`
Transporting files

Create on your system a simple text file 'example.txt':

- Ubuntu: use gedit
- Mac OS: use TextEdit
- Windows: use notepad

Install a file transfer program:

- Ubuntu: gftp sudo apt-get install gftp
- Mac OS: cyberduck from http://cyberduck.ch/

Use 'ssh2' or 'scp' as protocol

Click yourself a new directory on Lisa, and put 'example.txt' in that directory.

Try to download a file from Lisa to your system.
Tips about file names

- Do NOT use spaces: my file.txt → my-file.txt

- Use only these characters:
  
  abcdefghijklmnopqrstuvwxyz
  ABCDEFGHIJKLMNOPQRSTUVWXYZ
  0123456789
  _-

- Suffixes less important than in Windows, but it is wise to use .txt for simple text files, .jpg for jpeg files, etc.

- Upper and lower case do matter:
  Myfile != myfile
Some commands

bc calculator
cat concatenate
cd change directory
chomd change perm. bits
cp copy
eenv print environment
less view file
man manual
ls list subdirectory
mv rename
nano editor
pwd where am I

Redirection: > < >> |
Create simple text files

Methods:

- create a file on your system, and copy to Lisa
or
- create file on lisa using an editor, for example 'nano'

Type:

nano

GNU nano 2.2.4                New Buffer
Inspect files with less

Less is a browser for text files

"Less is more than more"

Example, type

```
env
```

and then

```
env | less
```

Commands in less:

- `q` quit
- `space` one page ahead
- `g` to start of file
- `G` to end of file
- `/text` search text
- `n` next occurrence
- `N` previous occurrence

... and many more
Wildcards

CD to the directory lisatutorial/scripts

type:

```
ls
ls -l p*
ls -l p*d
ls -l pr*
echo pr*
ls -l *.c
```

What is happening:

The shell is expanding `p*`, and presents the results to `ls`. `ls` sees these parameters:

```
-l parm  prod.c
```
Scripting

Exercise

Create the following file, and call it 'script'.

```bash
#!/bin/bash
echo "3 + 6" | bc > bcout
cp bcout result.txt
echo "contents of result.txt"
cat result.txt
```

A script is a kind of a workflow:
shell commands are executed.
Conditionals etc available.

Type:
```bash
chmod +x script
./script
```
Shell variables

Type:

v1=John

assign string "John" to shell variable v

echo $v1

three spaces

v2="John and Mary"

assign string with spaces to v

echo $v2

echo "$v2"

Notice the different results

example="a"
w=${example}b

Use {example} to prevent $exampleb

echo $w

Allowed characters in variable names: a-z A-Z 0-9 _
Environment variables

Create environment variable:

```bash
export ev=horse
or
ev=horse
export ev
```

Environment variables are copied to the environment of programs (scripts) you start. Shell variables are not copied.

Cd to `lisatutorial/scripts`

Type:

```bash
myvar=one
yourvar=two
export yourvar
cat envtest
./envtest
```
The PATH environment variable

Type:

echo $PATH
you get something like:
/sara/sw/modules-ng-64/wrappers/sara:/bin:/usr/bin:
    /usr/bin/X11:/usr/local/bin

The shell will search for programs in the directories:

/sara/sw/modules-ng-64/wrappers/sara
/bin
/usr/bin
/usr/bin/X11
/usr/local/bin

NOTE: no '. ' in PATH
The HOME environment variable

Type:

```
echo $HOME
```

you will get something like:

```
/home/wiltest
```

Type:

```
echo ~
```

Now it will be clear what the ' ~ ' in your prompt means.

Example: extend the PATH variable to search also for programs in $HOME/bin:

```
PATH=$HOME/bin:$PATH
```
Some more scripting

This example we already made:

```bash
#!/bin/bash

3 + 6 | bc > bcout

This script is written for the program `bash`, the shell. Other script eating programs exist: python, perl, ruby and many more

cp bcout result.txt

echo "contents of result.txt"
cat result.txt

This example we already made:

# !/bin/bash

$1 + $2 | bc > bcout

cp bcout result.txt

echo "contents of result.txt"
cat result.txt

And activate the script: ./script1 5 7

$1 will become the first parameter you give, $2 the second

copy this example (called 'script') to script1:

cp script script1

and edit script1 to become:

#!/bin/bash

$1 + $2 | bc > bcout

cp bcout result.txt

echo "contents of result.txt"
cat result.txt

$1 will become the first parameter you give, $2 the second
Thank you for listening!

**UNIX**

is user friendly

it's just selective about who its friends are