

# Working with linux

Survival guide for linux first time users in a computing lab

Tru Huynh

Bioinformatic Group  
Structural Biology and Chemistry Department  
Institut Pasteur, Paris, France

August 16th 2009 / Bioinformatics and Comparative  
Genome Analysis Course  
Hong Kong

# Purposes of this talk/workshop

Working with  
linux

Tru Huynh

Purposes

- teach/remind you the basic Unix commands to be actually learn something from this 2 weeks courses
- avoid painfull mistakes
- give you good/healthy Unix habits
- show you some other commands that might be helpfull sometime in the future
- NOT teaching you system administration

- teach/remind you the basic Unix commands to be actually learn something from this 2 weeks courses
- avoid painfull mistakes
- give you good/healthy Unix habits
- show you some other commands that might be helpfull sometime in the future
- NOT teaching you system administration

# Purposes of this talk/workshop

Working with  
linux

Tru Huynh

Purposes

- teach/remind you the basic Unix commands to be actually learn something from this 2 weeks courses
- avoid painfull mistakes
- give you good/healthy Unix habits
- show you some other commands that might be helpfull sometime in the future
- NOT teaching you system administration

# Purposes of this talk/workshop

Working with  
linux

Tru Huynh

Purposes

- teach/remind you the basic Unix commands to be actually learn something from this 2 weeks courses
- avoid painfull mistakes
- give you good/healthy Unix habits
- show you some other commands that might be helpfull sometime in the future
- NOT teaching you system administration

# Purposes of this talk/workshop

Working with  
linux

Tru Huynh

Purposes

- teach/remind you the basic Unix commands to be actually learn something from this 2 weeks courses
- avoid painfull mistakes
- give you good/healthy Unix habits
- show you some other commands that might be helpfull sometime in the future
- NOT teaching you system administration

- **1998 Graduate student ESPCI, Chemist**
- 2002 PhD in molecular modeling (folding mechanism of a multidomain protein), familiar with molecular dynamics simulation on small clusters to large computing center
- 2002 Join the Bioinformatic Group of P. M. Nilges at the Institut Pasteur as system admin (90 machines, 300 cores, 20 TB of shared storage/NFS, 20 users)  
<http://www.pasteur.fr>
- 2004 join the CentOS project <http://www.centos.org>

- 1998 Graduate student ESPCI, Chemist
- 2002 PhD in molecular modeling (folding mechanism of a multidomain protein), familiar with molecular dynamics simulation on small clusters to large computing center
- 2002 Join the Bioinformatic Group of P. M. Nilges at the Institut Pasteur as system admin (90 machines, 300 cores, 20 TB of shared storage/NFS, 20 users)  
<http://www.pasteur.fr>
- 2004 join the CentOS project <http://www.centos.org>

- 1998 Graduate student ESPCI, Chemist
- 2002 PhD in molecular modeling (folding mechanism of a multidomain protein), familiar with molecular dynamics simulation on small clusters to large computing center
- 2002 Join the Bioinformatic Group of P. M. Nilges at the Institut Pasteur as system admin (90 machines, 300 cores, 20 TB of shared storage/NFS, 20 users)  
<http://www.pasteur.fr>
- 2004 join the CentOS project <http://www.centos.org>

- 1998 Graduate student ESPCI, Chemist
- 2002 PhD in molecular modeling (folding mechanism of a multidomain protein), familiar with molecular dynamics simulation on small clusters to large computing center
- 2002 Join the Bioinformatic Group of P. M. Nilges at the Institut Pasteur as system admin (90 machines, 300 cores, 20 TB of shared storage/NFS, 20 users)  
<http://www.pasteur.fr>
- 2004 join the CentOS project <http://www.centos.org>

**Working with  
linux**

Tru Huynh

**Basic level**

CLI/GUI

Multiple OS

Remote  
connection

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

Basic level knowledge of unix systems.

Working with  
linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote  
connection

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

## Account informations

- login
- password
- members of groups
- just 'numbers': uid, gid
- \$HOME directory

Working with  
linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote  
connection

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

## Account informations

- login
- password
- members of groups
- just 'numbers': uid, gid
- \$HOME directory

Working with  
linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote  
connection

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

## Account informations

- login
- password
- members of groups
- just 'numbers': uid, gid
- \$HOME directory

Working with  
linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote  
connection

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

## Account informations

- login
- password
- members of groups
- just 'numbers': uid, gid
- \$HOME directory

Working with  
linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote  
connection

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

## Account informations

- login
- password
- members of groups
- just 'numbers': uid, gid
- \$HOME directory

## Working with linux

Tru Huynh

### Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources
- storage space (disk: local or remote)
- superuser: root

## Working with linux

Tru Huynh

### Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources
- storage space (disk: local or remote)
- superuser: root

## Working with linux

Tru Huynh

### Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources
- storage space (disk: local or remote)
- superuser: root

- files, folders or directories

```
dirA
|- dirB
   |- dirC
   |  |- file1
   |  |- file2
   |  `-- file3
   `-- dirD
      |- fileA
      `-- fileB
```

- relative and absolute PATH:

RELATIVE PATH:	ABSOLUTE PATH:
/	/
- etc	- /etc
- home	- /home
- lib	- /lib
- sbin	- /sbin
- tmp	- /tmp
- usr	- /usr
- bin	- /usr/bin
`-- bash	`-- /usr/bin/bash
`-- local	`-- /usr/local
- bin	- /usr/local/bin
`-- lib	`-- /usr/local/lib
`-- var	`-- /var

## Interaction through the keyboard ONLY ...

- the SHELL: interpret the words (space separated) you TYPE ...

- the case is IMPORTANT

```
ls LS lS Ls
```

are all different commands

- which shell? bash (default shell on linux) tcsh or zsh
- What is a command?

```
username@servername > programname -option1 ... argument1 ...
tru@callab > ls -l -d /home
```

## Interaction through the keyboard ONLY ...

- the SHELL: interpret the words (space separated) you TYPE ...

- the case is IMPORTANT

```
ls LS lS lS
```

are all different commands

- which shell? bash (default shell on linux) tcsh or zsh
- What is a command?

```
username@servername > programname -option1 ... argument1 ...
tru@callab > ls -l -d /home
```

## Interaction through the keyboard ONLY ...

- the SHELL: interpret the words (space separated) you TYPE ...

- the case is IMPORTANT

```
ls LS lS lS
```

are all different commands

- which shell? bash (default shell on linux) tcsh or zsh
- What is a command?

```
username@servername > programname -option1 ... argument1 ...
tru@callab > ls -l -d /home
```

## Interaction through the keyboard ONLY ...

- the SHELL: interpret the words (space separated) you TYPE ...

- the case is IMPORTANT

```
ls LS lS lS
```

are all different commands

- which shell? bash (default shell on linux) tcsh or zsh
- What is a command?

```
username@servername > programname -option1 ... argument1 ...
tru@callab > ls -l -d /home
```

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- historical: Xerox has invented the X windows and the mouse
- Apple/Macintosh made the mouse popular (one button), drag/drop
- Microsoft windows(TM) made it popular (cut/paste with ctrl-c/v) for all the others
- Linux/Unix GUI: highlight with the left button pressed, paste text in X windows -> middle mouse

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- historical: Xerox has invented the X windows and the mouse
- Apple/Macintosh made the mouse popular (one button), drag/drop
- Microsoft windows(TM) made it popular (cut/paste with ctrl-c/v) for all the others
- Linux/Unix GUI: highlight with the left button pressed, paste text in X windows -> middle mouse

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- historical: Xerox has invented the X windows and the mouse
- Apple/Macintosh made the mouse popular (one button), drag/drop
- Microsoft windows(TM) made it popular (cut/paste with ctrl-c/v) for all the others
- Linux/Unix GUI: highlight with the left button pressed, paste text in X windows -> middle mouse

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- historical: Xerox has invented the X windows and the mouse
- Apple/Macintosh made the mouse popular (one button), drag/drop
- Microsoft windows(TM) made it popular (cut/paste with ctrl-c/v) for all the others
- Linux/Unix GUI: highlight with the left button pressed, paste text in X windows -> middle mouse

## Working with linux

Tru Huynh

Basic level

CLI/GUI

**Multiple OS**

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- **PC under Windows**
- Commercial Unix (proprietary hardware): AIX, HPUX, Solaris, and dead now: IRIX(SGI), Tru64(Dec/Compaq/HP)
- Opensource Unix (MIT license/commodity hardware): \*BSD, OpenSolaris
- Opensource Unix (GPL/commoditu hardware): Linux (free or commercially supported)
- MacOSX (proprietary hardware with based on commodity hardware)
- but now standard to intel compatible

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- PC under Windows
- Commercial Unix (proprietary hardware): AIX, HPUX, Solaris, and dead now: IRIX(SGI), Tru64(Dec/Compaq/HP)
- Opensource Unix (MIT license/commodity hardware): \*BSD, OpenSolaris
- Opensource Unix (GPL/commoditu hardware): Linux (free or commercially supported)
- MacOSX (proprietary hardware with based on commodity hardware)
- but now standard to intel compatible

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- PC under Windows
- Commercial Unix (proprietary hardware): AIX, HPUX, Solaris, and dead now: IRIX(SGI), Tru64(Dec/Compaq/HP)
- Opensource Unix (MIT license/commodity hardware): \*BSD, OpenSolaris
- Opensource Unix (GPL/commoditu hardware): Linux (free or commercially supported)
- MacOSX (proprietary hardware with based on commodity hardware)
- but now standard to intel compatible

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- PC under Windows
- Commercial Unix (proprietary hardware): AIX, HPUX, Solaris, and dead now: IRIX(SGI), Tru64(Dec/Compaq/HP)
- Opensource Unix (MIT license/commodity hardware): \*BSD, OpenSolaris
- Opensource Unix (GPL/commoditu hardware): Linux (free or commercially supported)
- MacOSX (proprietary hardware with based on commodity hardware)
- but now standard to intel compatible

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- PC under Windows
- Commercial Unix (proprietary hardware): AIX, HPUX, Solaris, and dead now: IRIX(SGI), Tru64(Dec/Compaq/HP)
- Opensource Unix (MIT license/commodity hardware): \*BSD, OpenSolaris
- Opensource Unix (GPL/commoditu hardware): Linux (free or commercially supported)
- MacOSX (proprietary hardware with based on commodity hardware)
- but now standard to intel compatible

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- PC under Windows
- Commercial Unix (proprietary hardware): AIX, HPUX, Solaris, and dead now: IRIX(SGI), Tru64(Dec/Compaq/HP)
- Opensource Unix (MIT license/commodity hardware): \*BSD, OpenSolaris
- Opensource Unix (GPL/commoditu hardware): Linux (free or commercially supported)
- MacOSX (proprietary hardware with based on commodity hardware)
- but now standard to intel compatible

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

## What next?

- multiple boot the same hardware
- virtualisation (xen, vmware, virtualbox, ... )  
jail/zones/chroot
- same hardware/different OS/same time or not
- different hardware, but same OS: see NetBSD for example  
<http://www.netbsd.org/Ports/>: 57 supported ports
  - standard PC 32/64 bits
  - sparcs
  - alpha, mips, powerpc
  - ...
- multi arches software (firefox ,Acrobat Reader ,OpenOffice, skype, ... python, perl,..)

# Connecting to another machine (download/upload/remote shell)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

**Remote  
connection**

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

- **N to 1: client(s) to 1 server**
- different protocols
  - ftp:// (use ftp or better wget and lftp)
  - http:// (regular browser, wget, elinks lftp)
  - ssh
- any N to any M: peer to peer (torrent)

# Connecting to another machine (download/upload/remote shell)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

**Remote  
connection**

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

- N to 1: client(s) to 1 server
- different protocols
  - ftp:// (use ftp or better wget and lftp)
  - http:// (regular browser, wget, elinks lftp)
  - ssh
- any N to any M: peer to peer (torrent)

# Connecting to another machine (download/upload/remote shell)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

**Remote  
connection**

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

- N to 1: client(s) to 1 server
- different protocols
  - ftp:// (use ftp or better wget and lftp)
  - http:// (regular browser, wget, elinks lftp)
  - ssh
- any N to any M: peer to peer (torrent)

# Connecting to another machine (download/upload/remote shell)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

**Remote  
connection**

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

- N to 1: client(s) to 1 server
- different protocols
  - ftp:// (use ftp or better wget and lftp)
  - http:// (regular browser, wget, elinks lftp)
  - ssh
- any N to any M: peer to peer (torrent)

# Connecting to another machine (download/upload/remote shell)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

**Remote  
connection**

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

- N to 1: client(s) to 1 server
- different protocols
  - ftp:// (use ftp or better wget and lftp)
  - http:// (regular browser, wget, elinks lftp)
  - ssh
- any N to any M: peer to peer (torrent)

# Connecting to another machine (download/upload/remote shell)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

**Remote  
connection**

First  
commands

Regulars  
expressions

Moving  
around in the  
filesystem

Rights and  
permissions

Organise  
your project

Self help

- N to 1: client(s) to 1 server
- different protocols
  - ftp:// (use ftp or better wget and lftp)
  - http:// (regular browser, wget, elinks lftp)
  - ssh
- any N to any M: peer to peer (torrent)

- **whoami: display the user name**
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- **whoami**: display the user name
- **id**: display the user identity (login,uid, group(s), gid)
  - **date**: display the date, timezone
  - **hostname**: display the name of the server
  - **who, w, uptime**: who is logged? what they are doing, how long the system has been running?
  - **passwd**: change your password
  - **top**: interactive display sorted processes running
  - **firefox**: your usual web browser client
  - **ctrl-c**: tell current the program to stop
  - **&, ctrl-z, bg, fg**: job control
  - **alias**: display/create command lines shortcuts

- **whoami**: display the user name
- **id**: display the user identity (login,uid, group(s), gid)
- **date**: display the date, timezone
- **hostname**: display the name of the server
- **who, w, uptime**: who is logged? what they are doing, how long the system has been running?
- **passwd**: change your password
- **top**: interactive display sorted processes running
- **firefox**: your usual web browser client
- **ctrl-c**: tell current the program to stop
- **&, ctrl-z, bg, fg**: job control
- **alias**: display/create command lines shortcuts

- **whoami**: display the user name
- **id**: display the user identity (login,uid, group(s), gid)
- **date**: display the date, timezone
- **hostname**: display the name of the server
- **who, w, uptime**: who is logged? what they are doing, how long the system has been running?
- **passwd**: change your password
- **top**: interactive display sorted processes running
- **firefox**: your usual web browser client
- **ctrl-c**: tell current the program to stop
- **&, ctrl-z, bg, fg**: job control
- **alias**: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

- whoami: display the user name
- id: display the user identity (login,uid, group(s), gid)
- date: display the date, timezone
- hostname: display the name of the server
- who, w, uptime: who is logged? what they are doing, how long the system has been running?
- passwd: change your password
- top: interactive display sorted processes running
- firefox: your usual web browser client
- ctrl-c: tell current the program to stop
- &, ctrl-z, bg, fg: job control
- alias: display/create command lines shortcuts

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

**Regulars expressions**

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- **A[abz]:** A and one character (a or b or z)
- A[1-9]: A and one digit from 1 to 9
- \* : a chain of any length with any character
- \*.txt : a chain of any length with any character terminated by .txt
- \*[Aa]\*.txt : a chain of any length with any character containing an upper/lower case A and terminated by .txt

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

**Regulars expressions**

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- **A[abz]:** A and one character (a or b or z)
- **A[1-9]:** A and one digit from 1 to 9
- \* : a chain of any lenght with any character
- \*.txt : a chain of any lenght with any character terminated by .txt
- \*[Aa]\*.txt : a chain of any lenght with any character containing an upper/lower case A and terminated by .txt

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

**Regulars expressions**

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- `A[abz]`: A and one character (a or b or z)
- `A[1-9]`: A and one digit from 1 to 9
- `*` : a chain of any lenght with any character
- `*.txt` : a chain of any lenght with any character terminated by `.txt`
- `*[Aa]*.txt` : a chain of any lenght with any character containing an upper/lower case A and terminated by `.txt`

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

**Regulars expressions**

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- `A[abz]`: A and one character (a or b or z)
- `A[1-9]`: A and one digit from 1 to 9
- `*` : a chain of any lenght with any character
- `*.txt` : a chain of any lenght with any character terminated by `.txt`
- `*[Aa]*.txt` : a chain of any lenght with any character containing an upper/lower case A and terminated by `.txt`

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

**Regulars expressions**

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- `A[abz]`: A and one character (a or b or z)
- `A[1-9]`: A and one digit from 1 to 9
- `*` : a chain of any lenght with any character
- `*.txt` : a chain of any lenght with any character terminated by `.txt`
- `*[Aa]*.txt` : a chain of any lenght with any character containing an upper/lower case A and terminated by `.txt`

- **pwd**: print working directory
- `.` and `..` : the names of the current directory/the one above
- `cd`: change directory (move around the filesystem)
- `ls` (and options): list files/directories
- `file`: display the file type
- `mv`: move, rename files/directories
- `cp`: copy files/directories
- `rm`: remove files
- `mkdir`, `rmdir`: create/remove directories
- `cat`, `echo`: display the content of a file/variable
- `more`, `less`: pagers (type `q` to quit)
- `cmp`, `diff`: compare files

- **pwd**: print working directory
- **.** and **..** : the names of the current directory/the one above
  - **cd**: change directory (move around the filesystem)
  - **ls** (and options): list files/directories
  - **file**: display the file type
  - **mv**: move, rename files/directories
  - **cp**: copy files/directories
  - **rm**: remove files
  - **mkdir, rmdir**: create/remove directories
  - **cat, echo**: display the content of a file/variable
  - **more, less**: pagers (type q to quit)
  - **cmp, diff**: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
  - file: display the file type
  - mv: move, rename files/directories
  - cp: copy files/directories
  - rm: remove files
  - mkdir, rmdir: create/remove directories
  - cat, echo: display the content of a file/variable
  - more, less: pagers (type q to quit)
  - cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

- pwd: print working directory
- . and .. : the names of the current directory/the one above
- cd: change directory (move around the filesystem)
- ls (and options): list files/directories
- file: display the file type
- mv: move, rename files/directories
- cp: copy files/directories
- rm: remove files
- mkdir, rmdir: create/remove directories
- cat, echo: display the content of a file/variable
- more, less: pagers (type q to quit)
- cmp, diff: compare files

```

-rw-r--r--   1 tru      Bis           268 Nov 16 19:23 file1
drwxr-xr-x   2 tru      Bis           4096 Nov 16 19:23 dir2
uuugggooo
<-|  |->
 |  |
 |  |-----
 |  | rights/perms  r=4, w=2, x=1
 |  |-----
-----

```

- **ls -l**

- umask: display/set the default permissions
- chown: change the owner
- chgrp: change the group
- chmod: change the permission

```

-rw-r--r--   1 tru      Bis           268 Nov 16 19:23 file1
drwxr-xr-x   2 tru      Bis          4096 Nov 16 19:23 dir2
uuugggooo
<-|  |->
 |  |
 |  |-----
 |  | rights/perms  r=4, w=2, x=1
 |  |-----
-----

```

- **ls -l**
- **umask: display/set the default permissions**
- chown: change the owner
- chgrp: change the group
- chmod: change the permission

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

**Rights and permissions**

Organise your project

Self help

```

-rw-r--r--    1 tru      Bis           268 Nov 16 19:23 file1
drwxr-xr-x    2 tru      Bis           496 Nov 16 19:23 dir2
uuugggooo
<-|  |->
 |  |
 |  |-----
 |  | rights/perms  r=4, w=2, x=1
 |  |-----
-----

```

- ls -l
- umask: display/set the default permissions
- chown: change the owner
- chgrp: change the group
- chmod: change the permission

```

-rw-r--r--   1 tru      Bis           268 Nov 16 19:23 file1
drwxr-xr-x   2 tru      Bis          4096 Nov 16 19:23 dir2
uuugggooo
<-|  |->
 |  |
 |  |-----
 |  | rights/perms  r=4, w=2, x=1
 |  |-----
-----

```

- ls -l
- umask: display/set the default permissions
- chown: change the owner
- chgrp: change the group
- chmod: change the permission

```

-rw-r--r--   1 tru      Bis           268 Nov 16 19:23 file1
drwxr-xr-x   2 tru      Bis          4096 Nov 16 19:23 dir2
uuugggooo
<-|  |->
 |  |
 |  |-----
 |  | rights/perms  r=4, w=2, x=1
 |  |-----
-----

```

- ls -l
- umask: display/set the default permissions
- chown: change the owner
- chgrp: change the group
- chmod: change the permission

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
    - top: continuously display sorted processes
    - kill: send signals to process (stop, suspend, continue)
    - nice, renice: lower the priority
    - ulimit: display imposed limits
    - free: display the memory usage
  - storage
    - df: disk free
    - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- processor resources: top, ps, kill, nice, renice
  - ps: display processes
  - top: continuously display sorted processes
  - kill: send signals to process (stop, suspend, continue)
  - nice, renice: lower the priority
  - ulimit: display imposed limits
  - free: display the memory usage
- storage
  - df: disk free
  - du: disk used

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- meaningful naming (not lazy test, test1) but Analysis, journal/2009/08/16/log.txt
- easy to remember after 6 months, 3 years
- add "extensions" to files: log.txt, bday-2009.jpg not just name
- be consistent
- keep a daily log (cut paste command/output)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

**Organise your project**

Self help

- meaningful naming (not lazy test, test1) but Analysis, journal/2009/08/16/log.txt
- easy to remember after 6 months, 3 years
- add "extensions" to files: log.txt, bday-2009.jpg not just name
- be consistent
- keep a daily log (cut paste command/output)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

**Organise your project**

Self help

- meaningful naming (not lazy test, test1) but Analysis, journal/2009/08/16/log.txt
- easy to remember after 6 months, 3 years
- add "extensions" to files: log.txt, bday-2009.jpg not just name
- be consistent
- keep a daily log (cut paste command/output)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- meaningful naming (not lazy test, test1) but Analysis, journal/2009/08/16/log.txt
- easy to remember after 6 months, 3 years
- add "extensions" to files: log.txt, bday-2009.jpg not just name
- be consistent
- keep a daily log (cut paste command/output)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- meaningful naming (not lazy test, test1) but Analysis, journal/2009/08/16/log.txt
- easy to remember after 6 months, 3 years
- add "extensions" to files: log.txt, bday-2009.jpg not just name
- be consistent
- keep a daily log (cut paste command/output)

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

```

|- DATAMANIP
|- GCOMP
|  |- MLE
|  |  |- MLEseqnew
|  |  |  |- MCL
|  |  |  |  |- families
|  |  |  |  |  '- FAMILIES
|  |  |  |  |  '- multmatchmle
|  |  |  |  |  '- MTUseqnew
|  |  |  |  |  '- MULseqnew
|  |  |  |  |  '- MTU
|  |  |  |  |  '- MLEseqnew
|  |  |  |  |  '- MTUseqnew
.....
|- GENOMEDB
  |- MLE
  |- MTU
  |- MUL
  '- scripts

```

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- locate: find on the local disk(s)
- whatis: short info on program
- man (man man): display manual pages for a program
- find

```
[tru@callab BCGA2009]$ find . -name MUL
./GCOMP/MUL
./GENOMEDB/MUL
```

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- locate: find on the local disk(s)
- whatis: short info on program
- man (man man): display manual pages for a program
- find

```
[tru@callab BCGA2009]$ find . -name MUL
./GCOMP/MUL
./GENOMEDB/MUL
```

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- locate: find on the local disk(s)
- whatis: short info on program
- man (man man): display manual pages for a program
- find

```
[tru@callab BCGA2009]$ find . -name MUL
./GCOMP/MUL
./GENOMEDB/MUL
```

## Working with linux

Tru Huynh

Basic level

CLI/GUI

Multiple OS

Remote connection

First commands

Regulars expressions

Moving around in the filesystem

Rights and permissions

Organise your project

Self help

- locate: find on the local disk(s)
- whatis: short info on program
- man (man man): display manual pages for a program
- find

```
[tru@callab BCGA2009]$ find . -name MUL
./GCOMP/MUL
./GENOMEDB/MUL
```

Working with  
linux

Tru Huynh

Avoid typing

Avoid  
re-typing

Redirections

Manage your  
disk space

Work anytime

Do not  
reinvent the  
wheel

Intermediate level knowledge of unix systems.

- shell completion (tab key)

- alias, functions

```
alias
```

```
alias ll='ls -l'
```

```
alias grep='grep -color=tty -d skip'
```

- adding directories to PATH

- shell completion (tab key)

- alias, functions

```
alias
```

```
alias ll='ls -l'
```

```
alias grep='grep -color=tty -d skip'
```

- adding directories to PATH

- shell completion (tab key)
- alias, functions

```
alias
```

```
alias ll='ls -l'
```

```
alias grep='grep -color=tty -d skip'
```

- adding directories to PATH

```
PATH=$HOME/build/bin:$PATH export PATH
```

- **history (recall/modify the previous commands)**
- shell completion (arrow keys)
- shell scripts (loops, tests, ...) - scripting instead of doing it manually

- history (recall/modify the previous commands)
- shell completion (arrow keys)
- shell scripts (loops, tests, ...) - scripting instead of doing it manually

- history (recall/modify the previous commands)
- shell completion (arrow keys)
- shell scripts (loops, tests, ...) - scripting instead of doing it manually

## stdout and stdin redirections

- `> filename`

*"create and add to" filename*

```
program > filename
```

*the OUTPUT file is created BEFORE anything else happens.*

- `>> filename`

*"append to" filename*

- `< filename`

*"take filename as stdin"*

- `<< EOF`

*"take all as stdin until the EOF marker"*

## stdout and stdin redirections

- `> filename`  
*"create and add to" filename*
- `>> filename`  
*"append to" filename*  
`program » filename`
- `< filename`  
*"take filename as stdin"*
- `<< EOF`  
*"take all as stdin until the EOF marker"*

## stdout and stdin redirections

- `> filename`  
*"create and add to" filename*
- `>> filename`  
*"append to" filename*
- `< filename`  
*"take filename as stdin"*  
`program < filename`
- `<< EOF`  
*"take all as stdin until the EOF marker"*

## stdout and stdin redirections

- `> filename`  
*"create and add to" filename*
- `>> filename`  
*"append to" filename*
- `< filename`  
*"take filename as stdin"*
- `<< EOF`  
*"take all as stdin until the EOF marker"*

```
program << EOF
1st command
2nd command
EOF
```

## Chaining commands

- output/input with a PIPE: |  
    program1 | program2
- one after the other with ;
- &&  
    and
- or ||

## Chaining commands

- output/input with a PIPE: |
- one after the other with ;  
`program1; program2`
- &&  
and
- or ||

## Chaining commands

- output/input with a PIPE: |
- one after the other with ;
- &&  
and  

```
program1 && program2
```
- or ||

## Chaining commands

- output/input with a PIPE: |
- one after the other with ;
- &&  
and
- or ||

```
program1 || program2
```

- **ln, ln -s : make hard links, soft links**
- tar: manipulate "tar" files (tape archives: convert directory into files and backward)
- compress/uncompress, gzip/gunzip, bzip2/bunzip2: compression programs
- zip/unzip, rar/unrar: same but for the windows world

## Working with linux

Tru Huynh

Avoid typing

Avoid re-typing

Redirections

Manage your disk space

Work anytime

Do not reinvent the wheel

- `ln`, `ln -s` : make hard links, soft links
- `tar`: manipulate "tar" files (tape archives: convert diretory into files and backward)
- `compress/uncompress`, `gzip/gunzip`, `bzip2/bunzip2`: compression programs
- `zip/unzip`, `rar/unrar`: same but for the windows world

- ln, ln -s : make hard links, soft links
- tar: manipulate "tar" files (tape archives: convert directory into files and backward)
- compress/uncompress, gzip/gunzip, bzip2/bunzip2: compression programs
- zip/unzip, rar/unrar: same but for the windows world

- ln, ln -s : make hard links, soft links
- tar: manipulate "tar" files (tape archives: convert directory into files and backward)
- compress/uncompress, gzip/gunzip, bzip2/bunzip2: compression programs
- zip/unzip, rar/unrar: same but for the windows world

## Working with linux

Tru Huynh

Avoid typing

Avoid re-typing

Redirections

Manage your disk space

**Work anytime**

Do not reinvent the wheel

- **at: one time job at specified date/time**
- cron: repeat jobs
- ssh keys instead of password

## Working with linux

Tru Huynh

Avoid typing

Avoid re-typing

Redirections

Manage your disk space

## Work anytime

Do not reinvent the wheel

- at: one time job at specified date/time
- cron: repeat jobs
- ssh keys instead of password

## Working with linux

Tru Huynh

Avoid typing

Avoid re-typing

Redirections

Manage your disk space

**Work anytime**

Do not reinvent the wheel

- at: one time job at specified date/time
- cron: repeat jobs
- ssh keys instead of password

- lot of usefull commands available:

`grep, head, tail, cut, paste, sed, sort, uniq,  
printf, tac, tr,`

and chaining these commands with `|` to avoid  
temporary files

- Comprehensive Perl Archive Network (CPAN):  
<http://www.cpan.org/>
- python's Package Index: <http://pypi.python.org/pypi>

- lot of usefull commands available:  
`grep, head, tail, cut, paste, sed, sort, uniq, printf, tac, tr,`  
and chaining these commands with `|` to avoid temporary files
- Comprehensive Perl Archive Network (CPAN):  
<http://www.cpan.org/>
- python's Package Index: <http://pypi.python.org/pypi>

- lot of usefull commands available:

`grep, head, tail, cut, paste, sed, sort, uniq,  
printf, tac, tr,`

and chaining these commands with `|` to avoid  
temporary files

- Comprehensive Perl Archive Network (CPAN):  
<http://www.cpan.org/>
- python's Package Index: <http://pypi.python.org/pypi>

Working with  
linux

Tru Huynh

Avoid typing

Avoid  
re-typing

Redirections

Manage your  
disk space

Work anytime

Do not  
reinvent the  
wheel

gedit (default text editor) and nedit are nice GUI editors, but sometime you don't/can't have them readily available

- nano: basic text editor
- vi (esc :w :q)
- emacs (ctrl-x ctrl-c)

Working with  
linux

Tru Huynh

Avoid typing

Avoid  
re-typing

Redirections

Manage your  
disk space

Work anytime

Do not  
reinvent the  
wheel

gedit (default text editor) and nedit are nice GUI editors, but sometime you don't/can't have them readily available

- nano: basic text editor
- vi (esc :w :q)
- emacs (ctrl-x ctrl-c)

Working with  
linux

Tru Huynh

Avoid typing

Avoid  
re-typing

Redirections

Manage your  
disk space

Work anytime

Do not  
reinvent the  
wheel

gedit (default text editor) and nedit are nice GUI editors, but sometime you don't/can't have them readily available

- nano: basic text editor
- vi (esc :w :q)
- emacs (ctrl-x ctrl-c)

Advanced level knowledge of unix systems.

## .ssh/config see also ssh\_config(5):

```
Host ssh
HostName ssh.pasteur.fr
    LocalForward 20001 sillage.bis.pasteur.fr:22
```

```
Host sillage
HostName localhost
    Port 20001
    HostKeyAlias localhost-sillage
```

```
Host *.hku.hk
    IdentityFile ~/.ssh/id_dsa-hk2009
```

## SOCKS tunnels:

```
ssh -D7070 outpost.somewhere.outhere
```

```
[tru@darwin ~]$ ssh callab38  
Enter passphrase for key '/home/tru/.ssh/id_dsa-hk2009':
```

- **ssh-agent is here to help!**
- avoid typing the password key to your ssh keys!
- no reason not to have password protected keys

```
[tru@darwin ~]$ ssh callab38  
Enter passphrase for key '/home/tru/.ssh/id_dsa-hk2009':
```

- ssh-agent is here to help!
- avoid typing the password key to your ssh keys!
- no reason not to have password protected keys

```
[tru@darwin ~]$ ssh callab38  
Enter passphrase for key '/home/tru/.ssh/id_dsa-hk2009':
```

- ssh-agent is here to help!
- avoid typing the password key to your ssh keys!
- no reason not to have password protected keys

```
[tru@darwin ]$ ssh callab38
Enter passphrase for key '/home/tru/.ssh/id_dsa-hk2009':
```

- ssh-agent is here to help!
- avoid typing the password key to your ssh keys!
- no reason not to have password protected keys

## example:

```
[tru@darwin ]$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-lyJiVx9482/agent.9482; export SSH_AUTH_SOCK;
SSH_AGENT_PID=9483; export SSH_AGENT_PID;
echo Agent pid 9483;
[tru@darwin ]$ ssh-add ~/.ssh/id_dsa-hk2009
Enter passphrase for /home/tru/.ssh/id_dsa-hk2009:
Identity added: /home/tru/.ssh/id_dsa-hk2009 (/home/tru/.ssh/id_dsa-hk2009)
[tru@darwin ]$ ssh callab38
Last login: Mon Aug 10 14:05:54 2009 from sillage.bis.pasteur.fr
...
[tru@darwin ]$ ssh-add -l
1024 b3:65:f8:31:80:de:4d:fb:54:57:22:7d:b0:6b:63:6d /home/tru/.ssh/id_dsa-hk2009
(DSA)
[tru@darwin ]$ ssh-add -d /home/tru/.ssh/id_dsa-hk2009
Identity removed: /home/tru/.ssh/id_dsa-hk2009 (/home/tru/.ssh/id_dsa-hk2009.pub)

[tru@darwin ]$ ssh-add -l
The agent has no identities.
```

- **fixed name**

```
TMPFILE=/tmp/mytmp.file
```

- PID dependant

```
TMPFILE=/tmp/mytmp.file.$$
```

- PID and hostname

```
TMPFILE=/tmp/mytmp.file.hostname.$$
```

- **let the system manage it!**

```
TMPFILE=mktmp
```

- fixed name

```
TMPFILE=/tmp/mytmp.file
```

- PID dependant

```
TMPFILE=/tmp/mytmp.file.$$
```

- PID and hostname

```
TMPFILE=/tmp/mytmp.file.hostname.$$
```

- let the system manage it!

```
TMPFILE=mktmp
```

- fixed name

```
TMPFILE=/tmp/mytmp.file
```

- PID dependant

```
TMPFILE=/tmp/mytmp.file.$$
```

- PID and hostname

```
TMPFILE=/tmp/mytmp.file.hostname.$$
```

- let the system manage it!

```
TMPFILE=mktmp
```

- fixed name

```
TMPFILE=/tmp/mytmp.file
```

- PID dependant

```
TMPFILE=/tmp/mytmp.file.$$
```

- PID and hostname

```
TMPFILE=/tmp/mytmp.file.hostname.$$
```

- let the system manage it!

```
TMPFILE=mktmp
```

## why not plain copy?

```
tru@callab $ scp -prv dir1 dir2 ... \  
my.other.login@another.machine:backups
```

## because rsync does it better!

- efficient: (incremental update)
- sparse file aware
- secure: uses ssh by default
- keep timestamps

## why not plain copy?

```
tru@callab $ scp -prv dir1 dir2 ... \  
my.other.login@another.machine:backups
```

## because rsync does it better!

- efficient: (incremental update)
- sparse file aware
- secure: uses ssh by default
- keep timestamps

## why not plain copy?

```
tru@callab $ scp -prv dir1 dir2 ... \  
my.other.login@another.machine:backups
```

## because rsync does it better!

- efficient: (incremental update)
- sparse file aware
- secure: uses ssh by default
- keep timestamps

## why not plain copy?

```
tru@callab $ scp -prv dir1 dir2 ... \  
my.other.login@another.machine:backups
```

## because rsync does it better!

- efficient: (incremental update)
- sparse file aware
- secure: uses ssh by default
- keep timestamps

## why not plain copy?

```
tru@callab $ scp -prv dir1 dir2 ... \  
my.other.login@another.machine:backups
```

## because rsync does it better!

- efficient: (incremental update)
- sparse file aware
- secure: uses ssh by default
- keep timestamps

## example:

```
tru@callab $ rsync -aPHsv dir1 dir2 ... \  
-exclude '*.swap' -exclude ... \  
my.other.login@another.machine:backups
```

## You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program EVEN after login OUT!
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program EVEN after login OUT!
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- **Why another program? What else do you need?**
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program **EVEN** after login **OUT!**
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program EVEN after login OUT!
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program EVEN after login OUT!
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program **EVEN** after login **OUT!**
    - cut and paste between screen's windows without mouse
    - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program **EVEN** after login **OUT!**
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program **EVEN** after login **OUT!**
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

You already know about:

- backgrounding &
- nohup
- Why another program? What else do you need?
  - recover from a lost connection!
  - attach/re-attach your shell from any other machine
  - interact with your program **EVEN** after login **OUT!**
  - cut and paste between screen's windows without mouse
  - scrollbar and logging
- Demo!

- screen arguments:

screen	start screen
screen -S myname	start "myname" screen session
screen -ls	list screen sessions
screen -r screenid	re-attache screenid session

- inside screen (control + A)

d	detach
c	create a new window
A	rename window
a	send a ctrl+a to the window
"	list all window
n, <space>	go to next window
?	help

- screen arguments:

screen	start screen
screen -S myname	start "myname" screen session
screen -ls	list screen sessions
screen -r screenid	re-attache screenid session

- inside screen (control + A)

d	detach
c	create a new window
A	rename window
a	send a ctrl+a to the window
"	list all window
n, <space>	go to next window
?	help

- manage the dependencies of your pipeline:  
do not redo un-needed work

```
tru@callab $ make
```

- example of a Makefile with all in one directory:

```
# define the source/target
SRC= $(wildcard *.inp)
DEST = $(subst .inp,.out,$(SRC) )
# first target is the default target
all: $(DEST)
# generic rule, .inp -> .out
# $< is the name of the source file
# $@ is the name of the target file
%.out: %.inp
<tab>echo processing $< to $@
```

- example of a Makefile for subdirectories:

```
SUBDIRS = foo bar baz
.PHONY: subdirs $(SUBDIRS)
subdirs: $(SUBDIRS)
$(SUBDIRS):
<tab>$(MAKE) -C $@
```

- run in parallel any number of independant jobs

```
tru@callab $ make -j 3
```

- manage the dependencies of your pipeline:  
do not redo un-needed work

```
tru@callab $ make
```

- example of a Makefile with all in one directory:

```
# define the source/target
SRC= $(wildcard *.inp)
DEST = $(subst .inp,.out,$(SRC) )
# first target is the default target
all: $(DEST)
# generic rule, .inp -> .out
# $< is the name of the source file
# $@ is the name of the target file
%.out: %.inp
<tab>echo processing $< to $@
```

- example of a Makefile for subdirectories:

```
SUBDIRS = foo bar baz
.PHONY: subdirs $(SUBDIRS)
subdirs: $(SUBDIRS)
$(SUBDIRS):
<tab>$(MAKE) -C $@
```

- run in parallel any number of independant jobs

```
tru@callab $ make -j 3
```

- manage the dependencies of your pipeline:  
do not redo un-needed work

```
tru@callab $ make
```

- example of a Makefile with all in one directory:

```
# define the source/target
SRC= $(wildcard *.inp)
DEST = $(subst .inp,.out,$(SRC) )
# first target is the default target
all: $(DEST)
# generic rule, .inp -> .out
# $< is the name of the source file
# $@ is the name of the target file
%.out: %.inp
<tab>echo processing $< to $@
```

- example of a Makefile for subdirectories:

```
SUBDIRS = foo bar baz
.PHONY: subdirs $(SUBDIRS)
subdirs: $(SUBDIRS)
$(SUBDIRS):
<tab>$(MAKE) -C $@
```

- run in parallel any number of independant jobs

```
tru@callab $ make -j 3
```

- manage the dependencies of your pipeline:  
do not redo un-needed work

```
tru@callab $ make
```

- example of a Makefile with all in one directory:

```
# define the source/target
SRC= $(wildcard *.inp)
DEST = $(subst .inp,.out,$(SRC) )
# first target is the default target
all: $(DEST)
# generic rule, .inp -> .out
# $< is the name of the source file
# $@ is the name of the target file
%.out: %.inp
<tab>echo processing $< to $@
```

- example of a Makefile for subdirectories:

```
SUBDIRS = foo bar baz
.PHONY: subdirs $(SUBDIRS)
subdirs: $(SUBDIRS)
$(SUBDIRS):
<tab>$(MAKE) -C $@
```

- run in parallel any number of independant jobs

```
tru@callab $ make -j 3
```

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

## Master your text editor

Advanced shell functions

...

- nedit: ctrl + left button for rectangular selection
- vim: see "Bytes of vim"
- emacs:

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

## Master your text editor

Advanced shell functions

...

- nedit: ctrl + left button for rectangular selection
- vim: see "Bytes of vim"
- emacs:

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

## Master your text editor

Advanced shell functions

...

- nedit: ctrl + left button for rectangular selection
- vim: see "Bytes of vim"
- emacs:

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

Master your text editor

Advanced shell functions

...

- bash: be more efficient (bash-completion)
- zsh: even more interactive tools, but takes time to learn

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

Master your text editor

Advanced shell functions

...

- bash: be more efficient (bash-completion)
- zsh: even more interactive tools, but takes time to learn

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

Master your text editor

Advanced shell functions

...

- **lftp (bookmarks + scripts) and curl (scripts++) for http:// ftp:// scriptable access**
- developing projects with others
  - rcs,cvs
  - subversion, mercurial, ...
- What's your favorite tool?

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

Master your text editor

Advanced shell functions

...

- lftp (bookmarks + scripts) and curl (scripts++) for http:// ftp:// scriptable access
- developping projects with others
  - rcs,cvs
  - subversion, mercurial, ...
- What's your favorite tool?

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

Master your text editor

Advanced shell functions

...

- lftp (bookmarks + scripts) and curl (scripts++) for http:// ftp:// scriptable access
- developping projects with others
  - rcs,cvs
  - subversion, mercurial, ...
- What's your favorite tool?

## Working with linux

Tru Huynh

ssh

Temporary files management

Copy your files efficiently

screen

Make

Master your text editor

Advanced shell functions

...

- lftp (bookmarks + scripts) and curl (scripts++) for http:// ftp:// scriptable access
- developping projects with others
  - rcs,cvs
  - subversion, mercurial, ...
- What's your favorite tool?

- lftp (bookmarks + scripts) and curl (scripts++) for http:// ftp:// scriptable access
- developping projects with others
  - rcs,cvs
  - subversion, mercurial, ...
- What's your favorite tool?