Linux, part 1

Original by Dr. Changjiang Zhang

Modified (shorten) by Dr. Jefferson Fong

Outline

(A) Introduction to Linux

(B) Basic Linux Tutorial

- There are three tasks, but you don’t have to submit them.
- However the material in this lecture will be covered in Test 2.

(A) Intro to Linux

- The most popular OS
- What is Unix?
- What is Linux?
- Why study Unix / Linux?

Which OS is Used in Most Personal Computer?

1 **Personal computer (个人计算机) include:**

Personal digital assistant, PDA/Pocket PC

Smart phone

Tablet

Laptop

Desktop

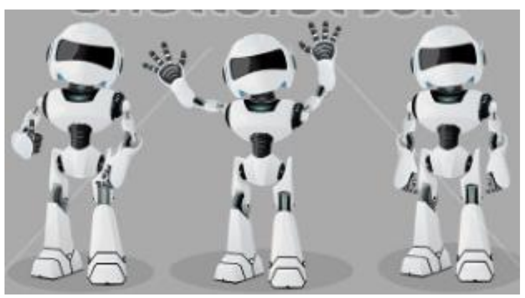
Which OS is Used in Embedded Machines?

Real-time operating systems (RTOS)

Embedded computers (嵌入式计算机) include:

Small specialized computers built into large components such as automobiles and appliances.

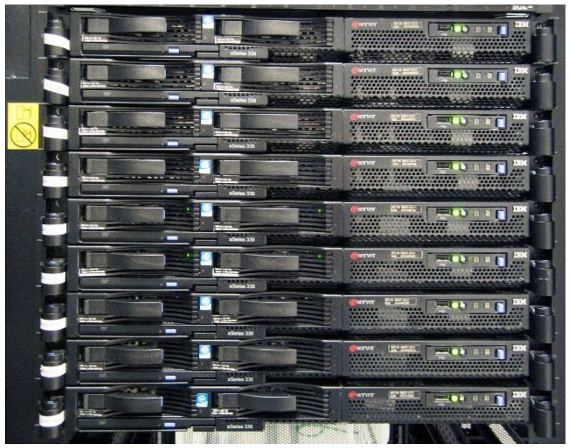
Chassis & Safety
Infotainment
Instrument Cluster
Gateways
Powertrain



Which OS is Used in Cluster Computers?

Rack mount servers (机架式服务器) or cluster computers

- E.g. servers for iSpace and UIC emails.



Which OS Used in Large or Supercomputers?

Supercomputer (parallel computing**):**
Large, powerful and **ultrafast** computer (similar to a mainframe, but much faster)

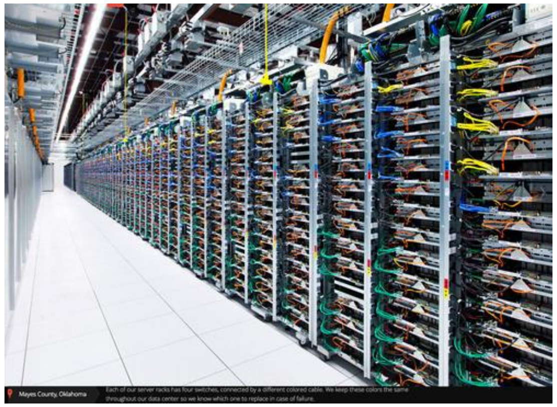
- IBM Blue GENE
- 天河2号
- 神威太湖之光
- Summit

Summit

Clusters (distributed computing)

- Google data centers
- Alibaba data centers
- Tencent data centers

Computers for **AI**, **Machine Learning** and **Deep Learning**



- The answer to all previous questions is

  Unix!

- One major deficiency of CST curriculum is that a student can graduate without learning Unix.
  - After you graduate and work in industries or attend grad school, if you tell your colleagues or boss you don’t know Unix, they would give you a funny look.
- In this course, we begin to remedy this problem.

“Linux is for Adults” – Stephan Grzesiek

What is Unix ?

- A multi-task and multi-user Operating System
- Developed in 1969 at AT&T's **Bell Labs** by
  - Ken Thompson (Unix)
  - Dennis Ritchie (C)
  - Douglas McIlroy (Pipes - Do one thing, do it well)
- Some other variants: System V, Solaris, SCO Unix, SunOS, 4.4BSD, FreeBSD, NetBSD, OpenBSD, BSDI

BSD: Berkeley Software Distribution

The old **Bell Labs Holmdel Complex in NJ**

What is Linux ?

- A **clone** of Unix
- Developed in 1991 by Linus Torvalds, a Finnish graduate student
- Consist of
  - Linux Kernel
  - GNU (GNU is Not Unix) Software
  - Software Package management
  - Other tools

What is Linux ?

Linux + GNU Utilities = Free Unix

Linux is an O/S **core** written by **Linus Torvalds** and **others**

- a set of small program tools written by **Richard Stallman** and others. They are the GNU utilities.

http://www.gnu.org/

- GNU is pronounced g'noo

What is Linux ?

- Originally developed for 32-bit x86-based PC
  - As an alternative for Windows OS
- Since then ported to other architectures, e.g.
  - Alpha, VAX, PowerPC, IBM S/390, MIPS, IA-64, ARM
  - PS2, TiVo, cellphones, watches, Nokia N810, NDS, routers, NAS, GPS, …

Linux Has Many Distributions

deepin™

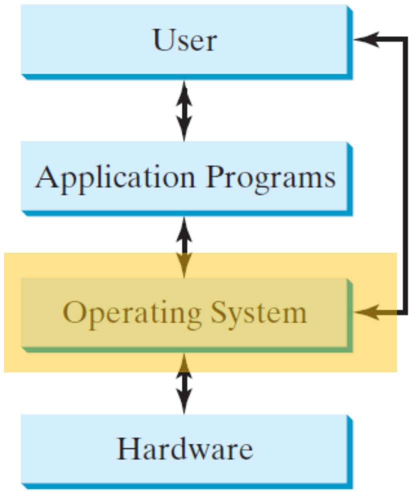
These are also Unix based:

- Apple Mac OS
- Huawei Harmony OS
- Google Android

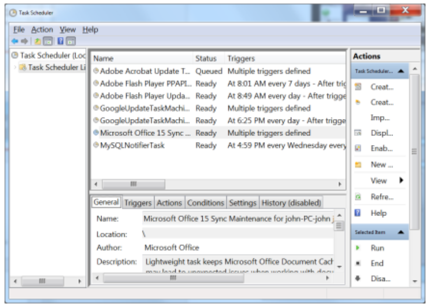
Why study Unix/Linux?

- Without Unix/Linux, there would be:
  - No internet (Most servers run on Unix/Linux)
  - No Modern Films (Most special effects are generated by Unix/Linux systems)
  - No Stocks and Bonds Sales (Almost all transactions are handled by Unix/Linux systems)
  - No ATMs or Banking (Most run Unix/Linux system)
  - No Electronic Games
  - No Military
  - No Operational Government
  - No Functioning Universities
  - No Large Corporations
  - …

A view of Operating System Services

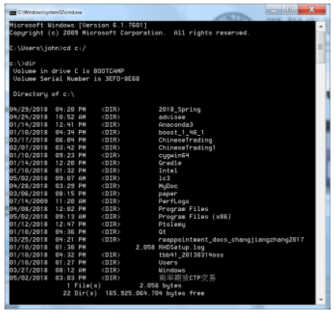


Empowers user to use OS directly.

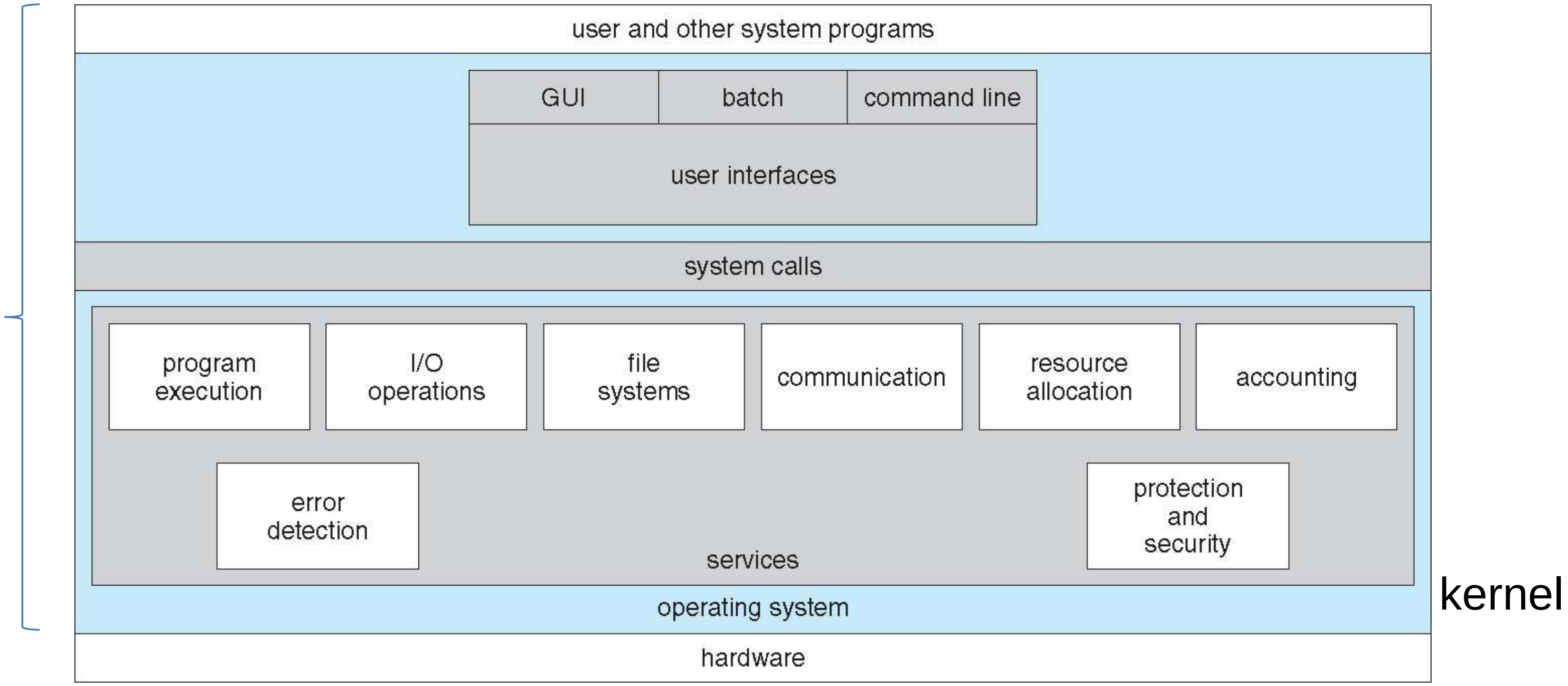


Name	Status	Triggers
Adobe Acrobat Update T...	Queued	Multiple triggers defined
Adobe Flash Player PPAPI...	Ready	At 8:01 AM every 7 days - After trig
Adobe Flash Player Upda...	Ready	At 8:49 AM every day - After trigge
GoogleUpdateTaskMachi...	Ready	Multiple triggers defined
GoogleUpdateTaskMachi...	Ready	At 6:25 PM every day - After trigge
Microsoft Office 15 Sync ...	Ready	Multiple triggers defined
MySQLNotifierTask	Ready	At 4:59 PM every Wednesday every

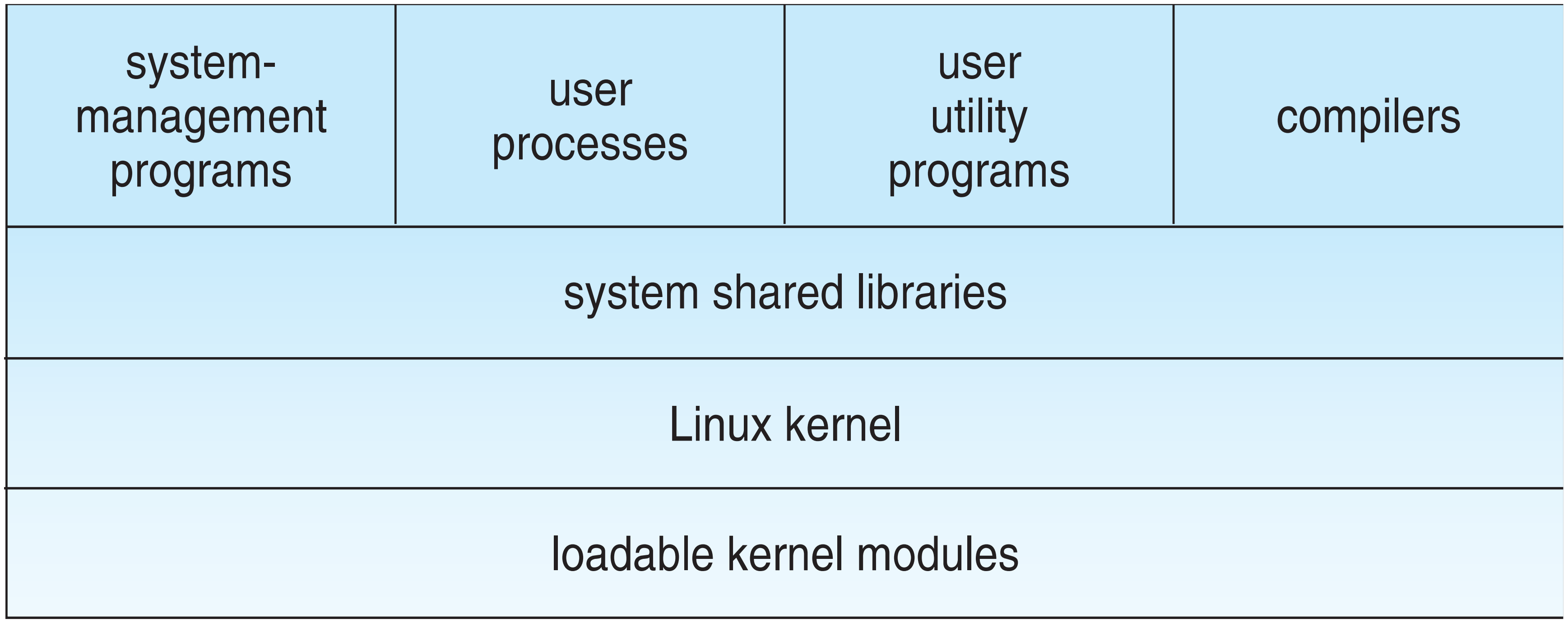
Name:	Microsoft Office 15 Sync Maintenance for john-PC-john
Location:	\
Author:	Microsoft Office
Description:	Lightweight task keeps Microsoft Office Document Cach



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\john>cd c:/

c:\>dir
 Volume in drive C is BOOTCAMP
 Volume Serial Number is 3EFD-8E68

 Directory of c:\

04/29/2018  04:28 PM    <DIR>          2018_Spring
04/24/2018  10:52 AM    <DIR>          advisee
01/14/2018  12:41 PM    <DIR>          Anaconda3
01/10/2018  04:34 PM    <DIR>          boost_1_46_1
03/17/2018  06:04 PM    <DIR>          ChineseTrading
02/07/2018  03:42 PM    <DIR>          ChineseTrading1
01/10/2018  09:23 PM    <DIR>          cygwin64
01/14/2018  12:20 PM    <DIR>          Gradle
01/10/2018  01:32 PM    <DIR>          Intel
05/02/2018  09:07 AM    <DIR>          lc3
04/28/2018  03:29 PM    <DIR>          MyDoc
03/06/2018  08:15 PM    <DIR>          paper
07/14/2009  11:20 AM    <DIR>          PerfLogs
04/08/2018  12:02 PM    <DIR>          Program Files
05/02/2018  09:13 AM    <DIR>          Program Files (x86)
01/12/2018  12:47 PM    <DIR>          Ptolemy
01/10/2018  04:36 PM    <DIR>          Qt
03/25/2018  04:21 PM    <DIR>          reappointment_docs_changjiangzhang2017
01/10/2018  01:30 PM             2,058 RHDSetup.log
01/10/2018  04:32 PM    <DIR>          tbb41_20130314oss
01/10/2018  01:27 PM    <DIR>          Users
03/27/2018  08:12 AM    <DIR>          Windows
05/02/2018  03:03 PM    <DIR>          南华期货CTP交易
               1 File(s)          2,058 bytes
              22 Dir(s)  165,925,064,704 bytes free
```



Components of Linux System



Linux Consists of Small Programs

Linux commands: “Each small program does one thing well”

Network: ssh, scp, ping, telnet, nslookup, wget

Shells: BASH, TCSH, alias, watch, clear, history, chsh, echo, set, setenv, xargs

System Information: w, whoami, man, info, which, free, echo, date, cal, **df**

Command Information: man, info

Symbols: |, >, >>, <, &, >&, 2>&1, ;, ~, ., .., $!, !:<n>, !<n>

Filters: grep, egrep, more, less, head, tail

Hotkeys: <ctrl><c>, <ctrl><d>

File System: ls, mkdir, cd, pwd, mv, ln, touch, cat, file, find, diff, cmp, /net/<hostname>/<path>, mount, du, **df**, chmod

Line Editors: awk, sed

File Editors: vi, vim, gvim, emacs –nw, emacs

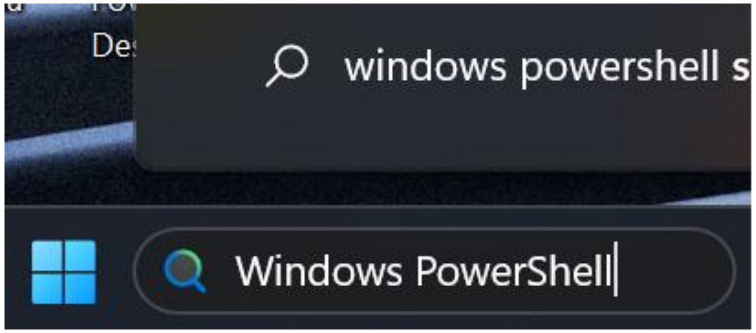
Process Management: ps, top, kill, killall, fg, bg

(B) Basic Linux Tutorial - Outline

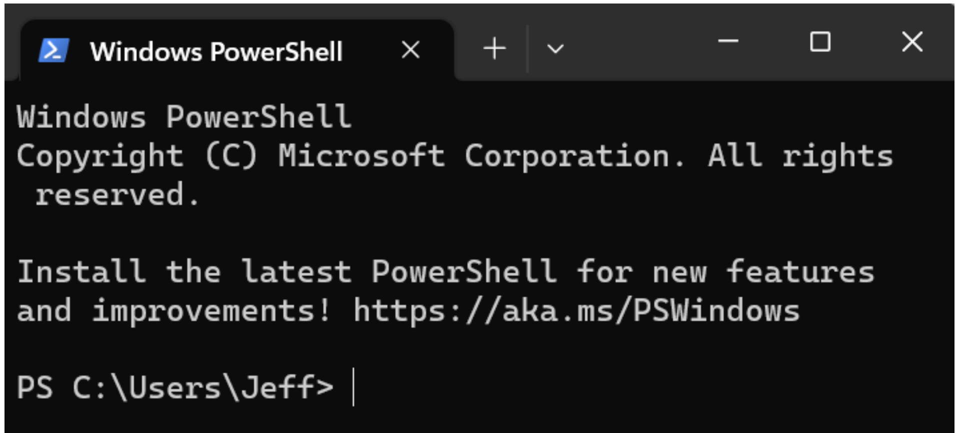
- Login to bcrab
- Basic Linux commands
  - Help, manual, pwd, ls, mkdir, rmdir, cd *, ls (more),
- Task1
- File commands
  - file, cp, mv, cat, more, less, head, tail, rm
- Task2
- ~~File permissions, chmod~~
- ~~Running jobs, ps, kill, top~~
- ~~Task3~~

Log in to stuweb.bcrab.cn

- For Mac users, use the Unix shell terminal in your computer instead.
- For Windows users, use PowerShell
  - At bottom of Windows screen, search for “Windows Powershell”



  - PowerShell should open.
  - You can also use Command Prompt.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights
 reserved.

Install the latest PowerShell for new features
and improvements! https://aka.ms/PSWindows

PS C:\Users\Jeff>
```

Log in to stuweb.bcrab.cn

- In PowerShell or Command Prompt, type

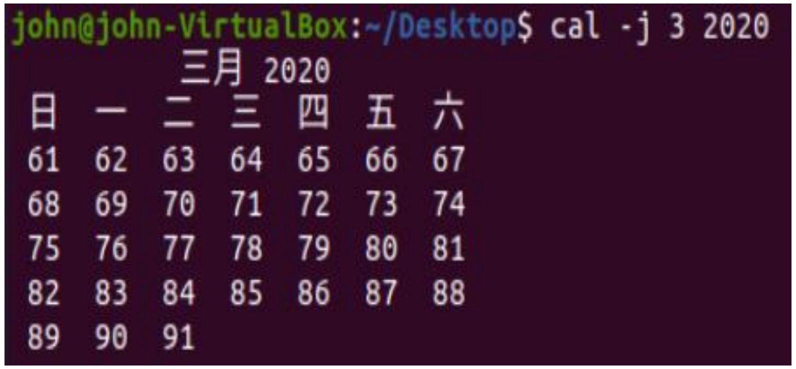
```
ssh account@stuweb.bcrab.cn
```

  - Change account to your bcrab account.
  - Enter your bcrab password when asked.
    - Note the cursor don’t move when you type in your password, so prying eye won’t know how many characters are in your password.

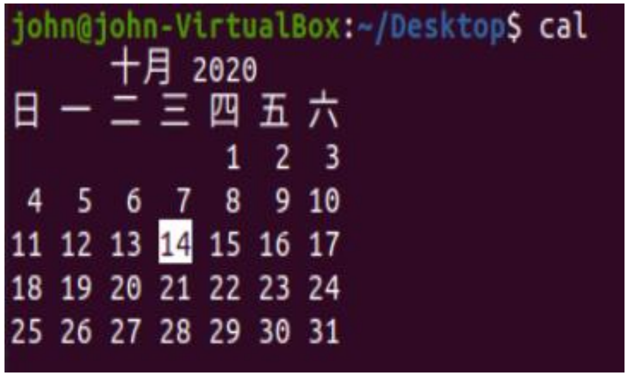
- In the following slides, type in the **Linux commands** in your computer.

Linux Command

- Command has three parts:
  - ***command*** *options* and *parameters*.
- Example:
  - **cal –j 3 2020**.
  - "cal" is the command,
  - "-j" is an option (Julian year)
  - "3" and "2020" are parameters (month and year)
  - **cal**



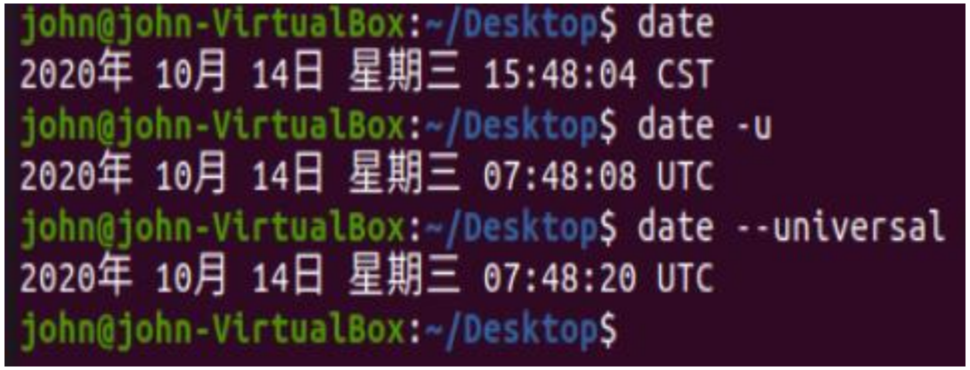
```
john@john-VirtualBox:~/Desktop$ cal -j 3 2020
            三月 2020
 日  一  二  三  四  五  六
 61  62  63  64  65  66  67
 68  69  70  71  72  73  74
 75  76  77  78  79  80  81
 82  83  84  85  86  87  88
 89  90  91
```



```
john@john-VirtualBox:~/Desktop$ cal
      十月 2020
日 一 二 三 四 五 六
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

Linux Command

- ❑ Command option has long and short forms.
- ❑ Example:
  - **date**
  - **date –u**
  - **date --universal**



```
john@john-VirtualBox:~/Desktop$ date
2020年 10月 14日 星期三 15:48:04 CST
john@john-VirtualBox:~/Desktop$ date -u
2020年 10月 14日 星期三 07:48:08 UTC
john@john-VirtualBox:~/Desktop$ date --universal
2020年 10月 14日 星期三 07:48:20 UTC
john@john-VirtualBox:~/Desktop$
```

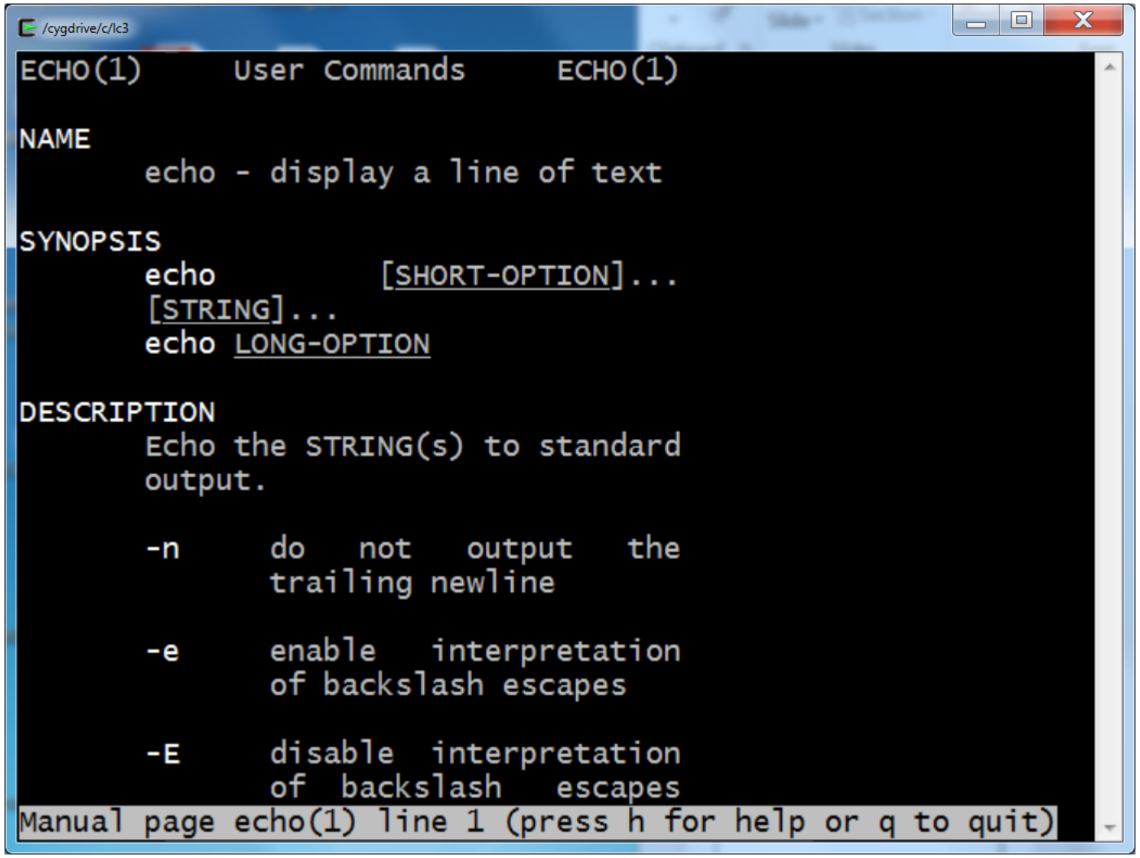
- The option –u is the short form, --universal is long form
- You can guess what these options mean, or search online for “Linux command date”

Help Manual

- Type "man" (for manual) and the command name. E.g. **man echo**

- Hit spacebar for next page
- Hit "q" to quit.

Can also search online for "Linux command echo"



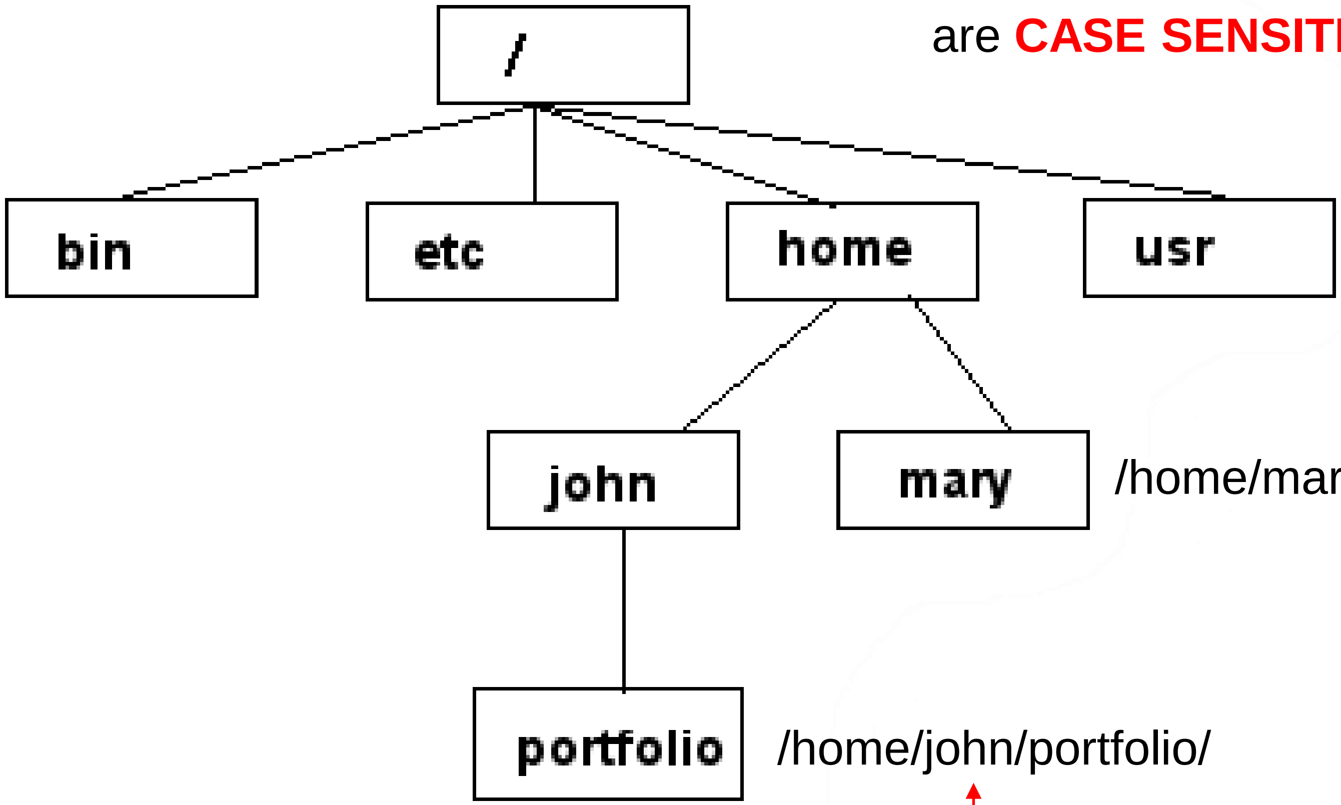
```
ECHO(1)      User Commands      ECHO(1)

NAME
       echo - display a line of text

SYNOPSIS
       echo          [SHORT-OPTION]...
       [STRING]...
       echo LONG-OPTION

DESCRIPTION
       Echo the STRING(s) to standard
       output.

       -n     do   not   output   the
              trailing newline

       -e     enable   interpretation
              of backslash escapes

       -E     disable  interpretation
              of  backslash   escapes
Manual page echo(1) line 1 (press h for help or q to quit)
```

Command History and Command Line Editing

- Use the up ↑ and down ↓ arrow keys to choose a previous command you've entered.
- Use left ← , right → arrow, <Del> and <Backspace>, or type in (insert) characters to make changes to that commend.
- After changing the command, hit enter to submit the new command.

Unix/Linux File System

NOTE: Unix file names are **CASE SENSITIVE!**



/home/mary/

/home/john/portfolio/

The Path

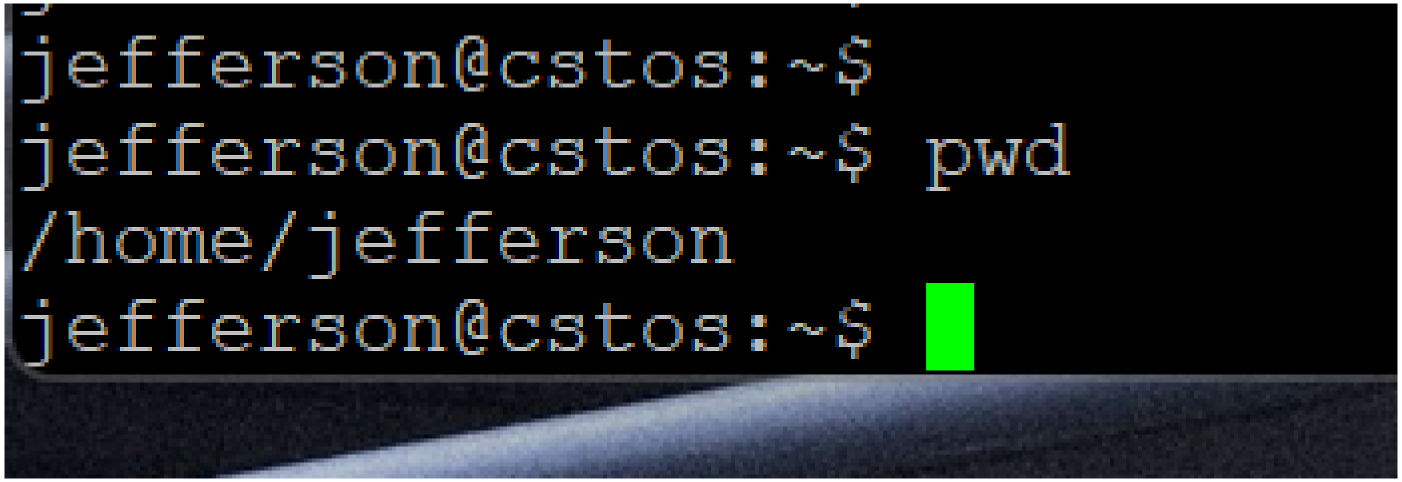
UNIX use forward slash,
Windows use backslash,

Some Unix and Command Prompt commands are similar.

- Original Microsoft DOS was an inferior copy of Unix.

Linux Command **pwd**

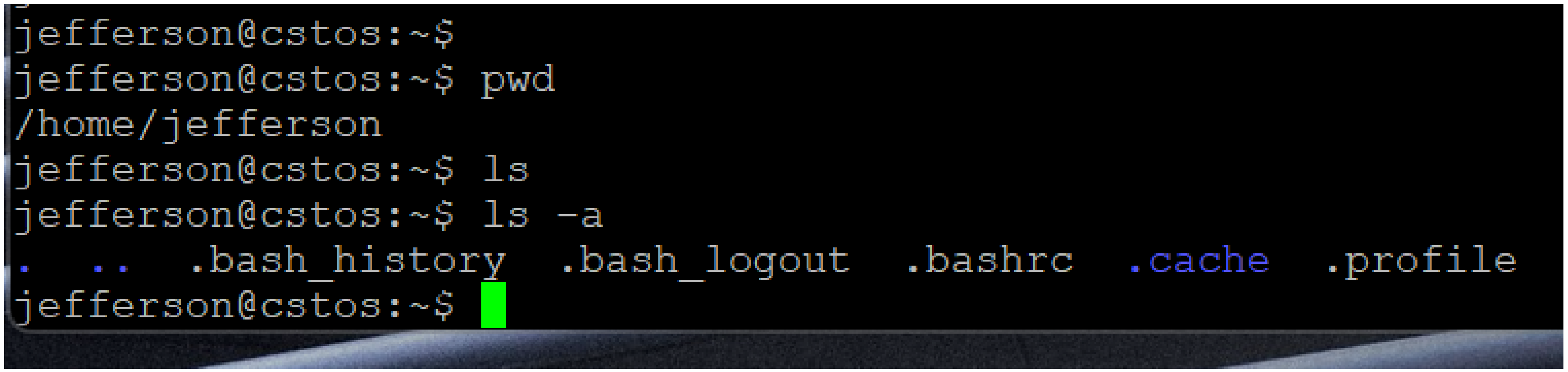
- Show the current path of working directory:
  - **pwd**
  - Shows current directory is /home/jefferson



```
jefferson@cstos:~$
jefferson@cstos:~$ pwd
/home/jefferson
jefferson@cstos:~$
```

Linux Command ls

- Type "ls" to list the contents in current directory.
- "ls" shows nothing in this directory; so try "ls –a" to list all.
- We see there are several hidden files with filenames beginning with "."



```
jefferson@cstos:~$
jefferson@cstos:~$ pwd
/home/jefferson
jefferson@cstos:~$ ls
jefferson@cstos:~$ ls -a
.   ..   .bash_history   .bash_logout   .bashrc   .cache   .profile
jefferson@cstos:~$
```

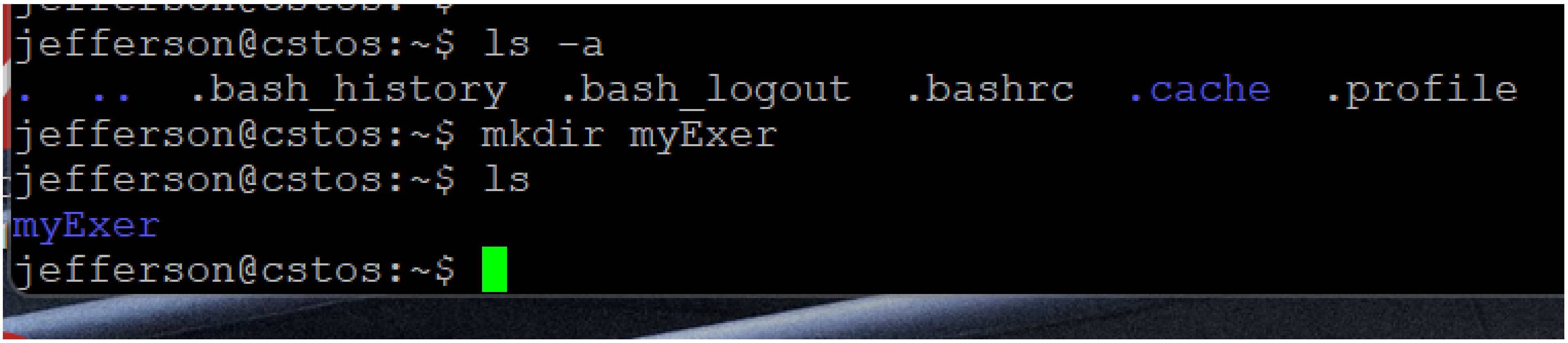
Command: mkdir

- Make (a new) directory

  mkdir *directory_name*

- E.g. **mkdir myExer**
- Type "ls" to see that myExer directory (in blue) has been created.

Same as Command Prompt in PC.



```
jefferson@cstos:~$ ls -a
.   ..   .bash_history  .bash_logout  .bashrc  .cache  .profile
jefferson@cstos:~$ mkdir myExer
jefferson@cstos:~$ ls
myExer
jefferson@cstos:~$
```

Linux Command **rmdir**

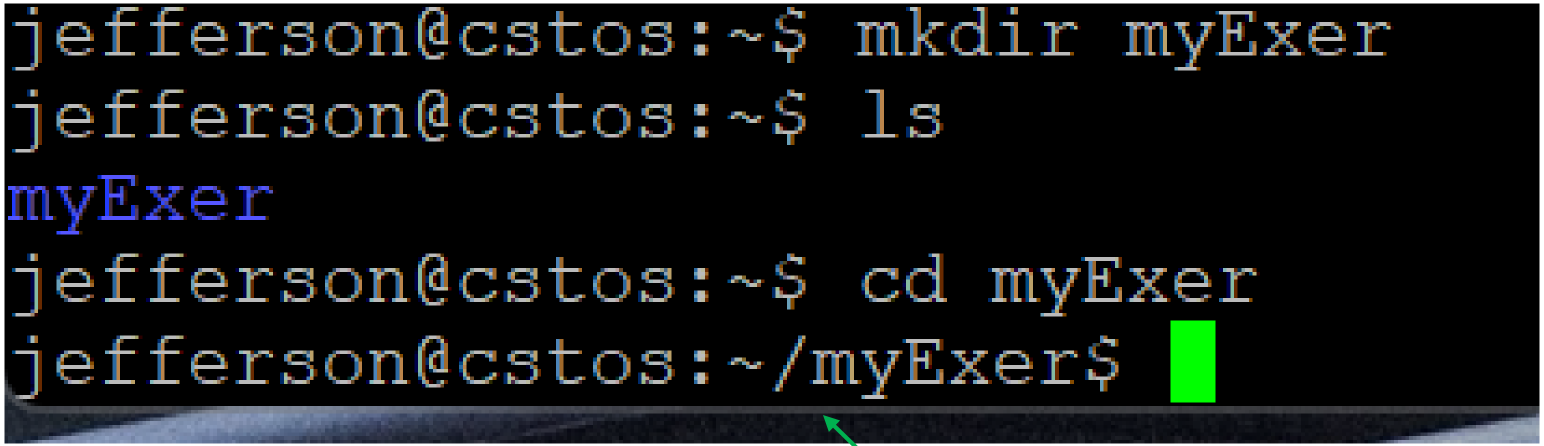
Same as Command Prompt in PC.

- Remove an empty directory
  - **rmdir *directory_name***
- If you made a mistake in creating the directory, you can remove it.
- Note: You can remove a nonempty directory using **rm**; more later.

Linux Command **cd**

- Change directory
  - **cd [directory_name]**
- E.g. **cd myExer**

Same as Command Prompt in PC.



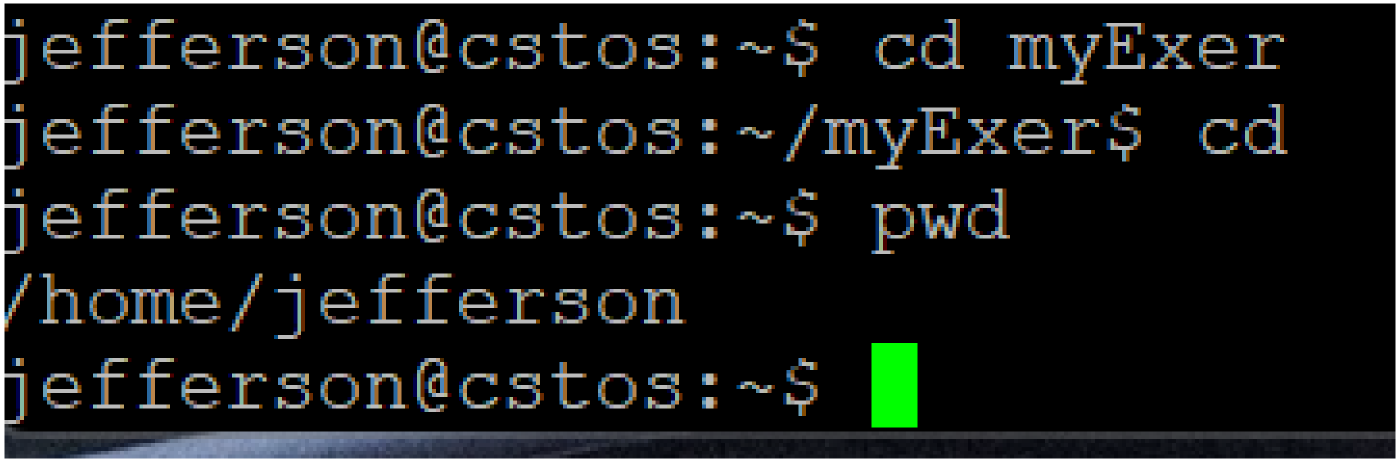
```
jefferson@cstos:~$ mkdir myExer
jefferson@cstos:~$ ls
myExer
jefferson@cstos:~$ cd myExer
jefferson@cstos:~/myExer$ 
```

- We are in the directory ~/myExer
- ~ means the user's home directory

Linux Command **cd**

- Change to home directory
  - **cd ~**
  - **cd** (also works in most systems)

“cd” is same as Command Prompt in PC.

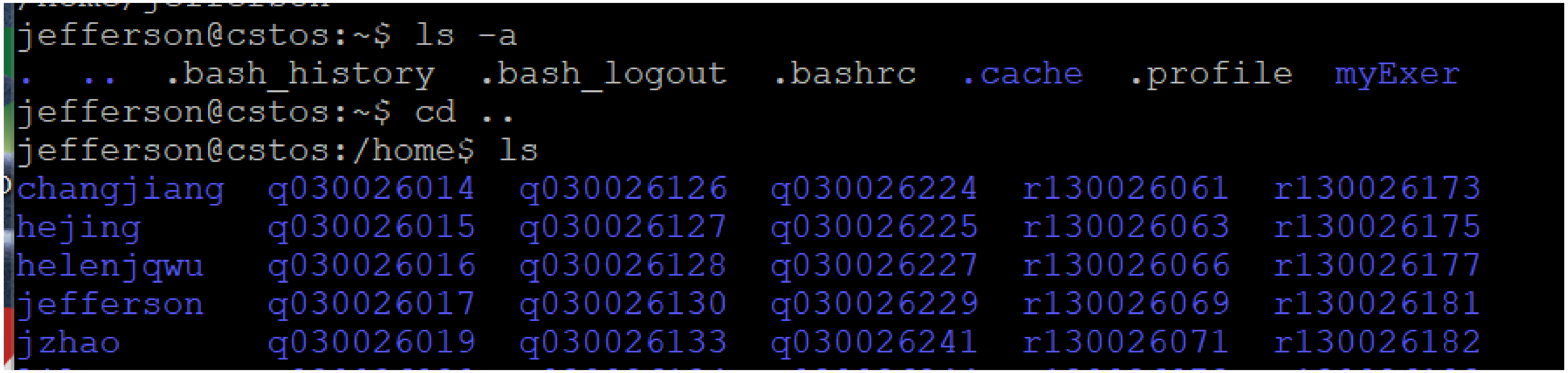


```
jefferson@cstos:~$ cd myExer
jefferson@cstos:~/myExer$ cd
jefferson@cstos:~$ pwd
/home/jefferson
jefferson@cstos:~$ 
```

Linux Command **cd**

- Change to parent directory (one level up).
  - **cd ..**

Same as Command Prompt in PC.

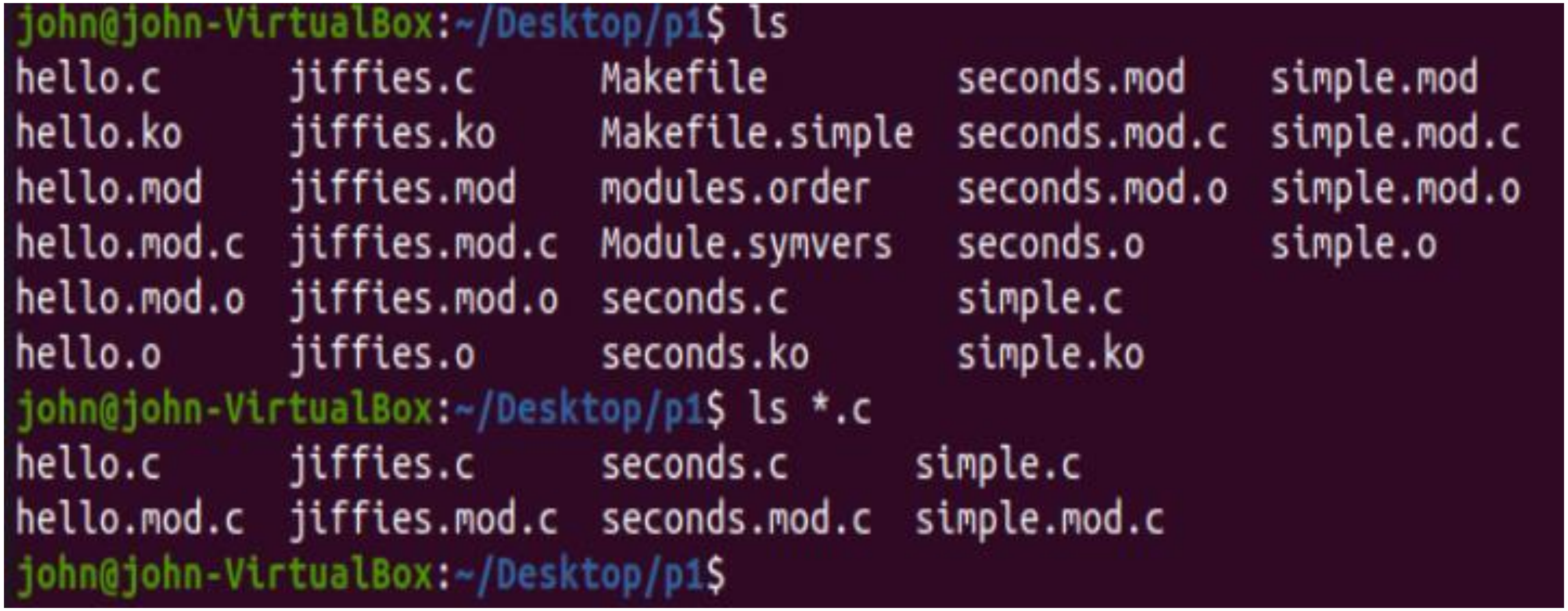


```
jefferson@cstos:~$ ls -a
.   ..   .bash_history   .bash_logout   .bashrc   .cache   .profile   myExer
jefferson@cstos:~$ cd ..
jefferson@cstos:/home$ ls
changjiang   q030026014   q030026126   q030026224   r130026061   r130026173
hejing       q030026015   q030026127   q030026225   r130026063   r130026175
helenjqwu    q030026016   q030026128   q030026227   r130026066   r130026177
jefferson    q030026017   q030026130   q030026229   r130026069   r130026181
jzhao        q030026019   q030026133   q030026241   r130026071   r130026182
```

Wildcard Character *

- Wildcard character *
- E.g. **ls *.c**
  - Lists all files with the form (something).c
  - i.e. list all C files in the directory.

Same as Command Prompt in PC.



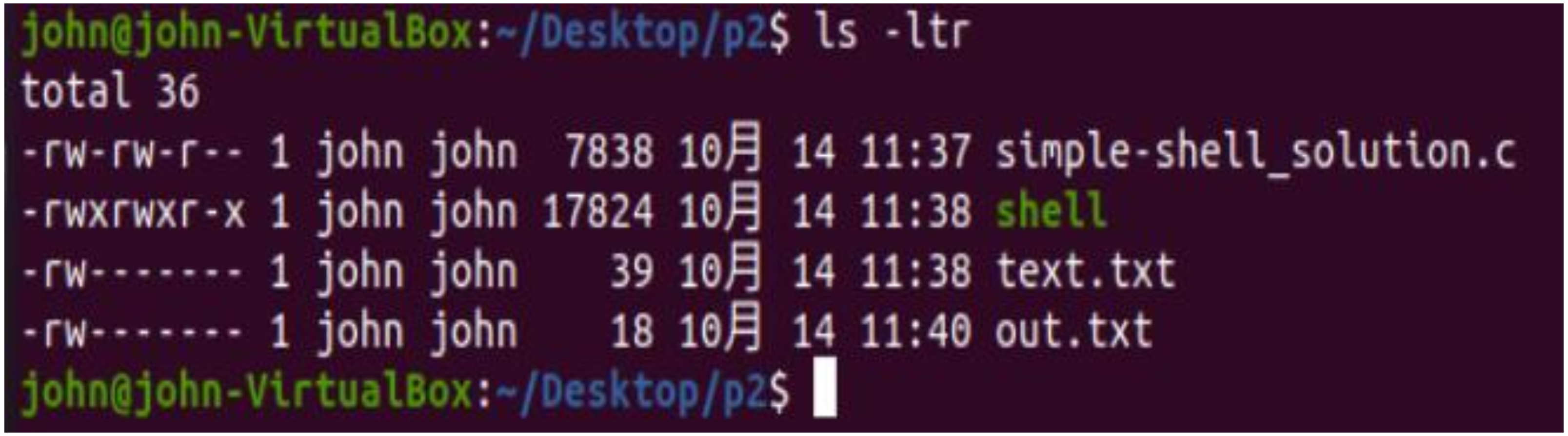
```
john@john-VirtualBox:~/Desktop/p1$ ls
hello.c      jiffies.c      Makefile          seconds.mod     simple.mod
hello.ko     jiffies.ko     Makefile.simple   seconds.mod.c   simple.mod.c
hello.mod    jiffies.mod    modules.order     seconds.mod.o   simple.mod.o
hello.mod.c  jiffies.mod.c  Module.symvers    seconds.o       simple.o
hello.mod.o  jiffies.mod.o  seconds.c         simple.c
hello.o      jiffies.o      seconds.ko        simple.ko
john@john-VirtualBox:~/Desktop/p1$ ls *.c
hello.c      jiffies.c      seconds.c      simple.c
hello.mod.c  jiffies.mod.c  seconds.mod.c  simple.mod.c
john@john-VirtualBox:~/Desktop/p1$
```

Command: ls

- ls has many options
  - ✓ -l lists files and folders with associated permissions
  - ✓ -t sort by modification time, newest first
  - ✓ -S sort by size
  - ✓ -h list file sizes in human-readable format
  - ✓ -r reverse the order
  - ✓ -a list all files including the dots and hidden files
  - ✓ -i print the index number (inode) of each file
  - ✓ -F List files and directories, with a forward slash (/) at the end of each directory
- See "**man ls**" for more options
- Options can be combined, e.g. "**ls -ltr**"

Linux Command **ls**

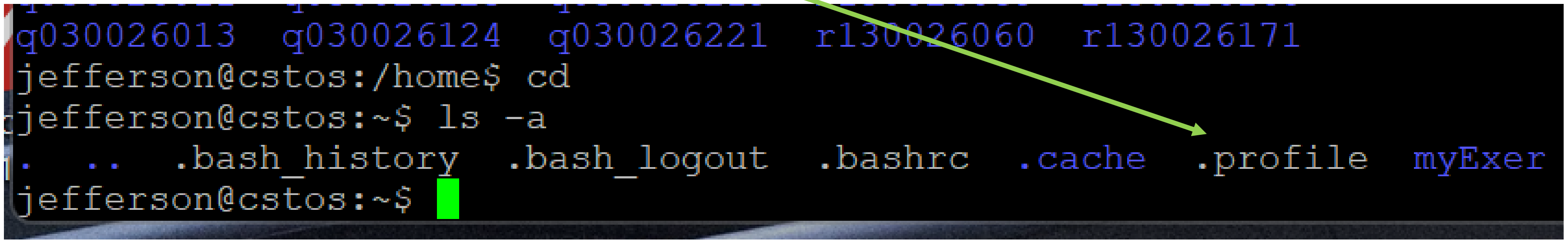
- List files by time in reverse order with long listing
  - **ls -ltr**



```
john@john-VirtualBox:~/Desktop/p2$ ls -ltr
total 36
-rw-rw-r-- 1 john john  7838 10月 14 11:37 simple-shell_solution.c
-rwxrwxr-x 1 john john 17824 10月 14 11:38 shell
-rw------- 1 john john    39 10月 14 11:38 text.txt
-rw------- 1 john john    18 10月 14 11:40 out.txt
john@john-VirtualBox:~/Desktop/p2$ 
```

Task 1

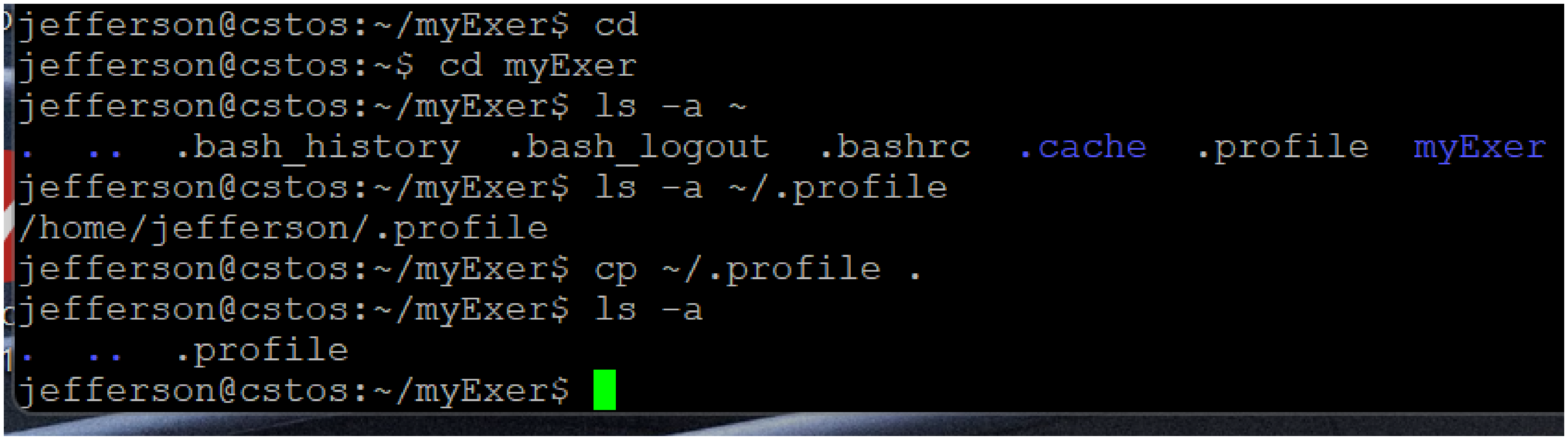
- Go to your home directory.
  - Use **pwd** and **cd**
- If you have not done so, make a directory called myExer
  - Use **mkdir**
- Lists the hidden files in your home directory
  - Note the file .profile



```
q030026013  q030026124  q030026221  r130026060  r130026171
jefferson@cstos:/home$ cd
jefferson@cstos:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .profile  myExer
jefferson@cstos:~$ 
```

Task 1

- Go to the myExer directory.
  - Use cd
- Copy the file .profile to this directory
  - See next slide for how to copy.



```
jefferson@cstos:~/myExer$ cd
jefferson@cstos:~$ cd myExer
jefferson@cstos:~/myExer$ ls -a ~
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .profile  myExer
jefferson@cstos:~/myExer$ ls -a ~/.profile
/home/jefferson/.profile
jefferson@cstos:~/myExer$ cp ~/.profile .
jefferson@cstos:~/myExer$ ls -a
.  ..  .profile
jefferson@cstos:~/myExer$ 
```

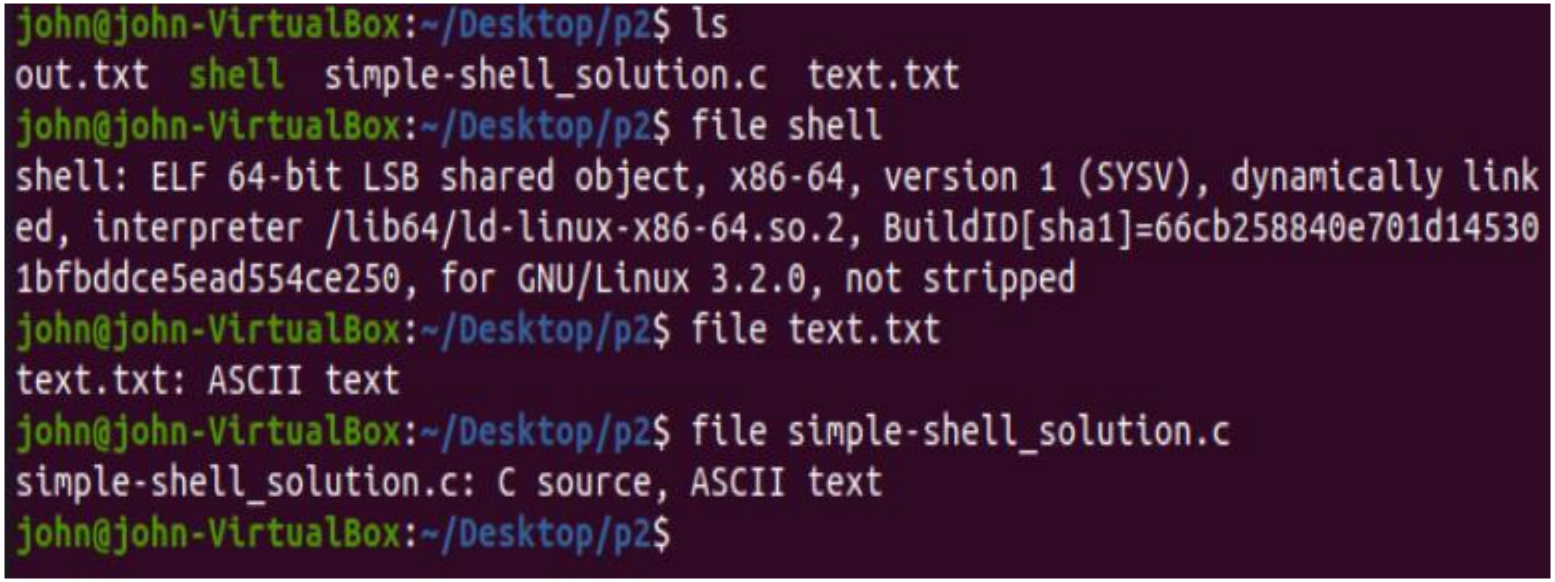
File Commands

- Copying a file: **cp**
- Move or rename a file: **mv**
- Remove/delete a file: **rm**
- Change file timestamp or create an empty new file: **touch**
- Sort the contents of the file filename: **sort**
- Remove duplicate adjacent lines from filename: **uniq**
- Determine file type: **file**

```
sort filename
uniq filename
touch  file1
file  filename
```

Linux Command file

- Show file type info:
  - **file *filename***



```
john@john-VirtualBox:~/Desktop/p2$ ls
out.txt  shell  simple-shell_solution.c  text.txt
john@john-VirtualBox:~/Desktop/p2$ file shell
shell: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically link
ed, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=66cb258840e701d14530
1bfbddce5ead554ce250, for GNU/Linux 3.2.0, not stripped
john@john-VirtualBox:~/Desktop/p2$ file text.txt
text.txt: ASCII text
john@john-VirtualBox:~/Desktop/p2$ file simple-shell_solution.c
simple-shell_solution.c: C source, ASCII text
john@john-VirtualBox:~/Desktop/p2$
```

File Command cp

- Copy files:
  - **cp**

- E.g. copy the file .profile from the home directory to the current directory.

```
cp ~/.profile .
```

- The last "." means the target file (in the current directory) will have the same name as the source directory (in the home directory)

File Command mv

- Move or rename a file:
  - **mv**

Move a file to its parent directory

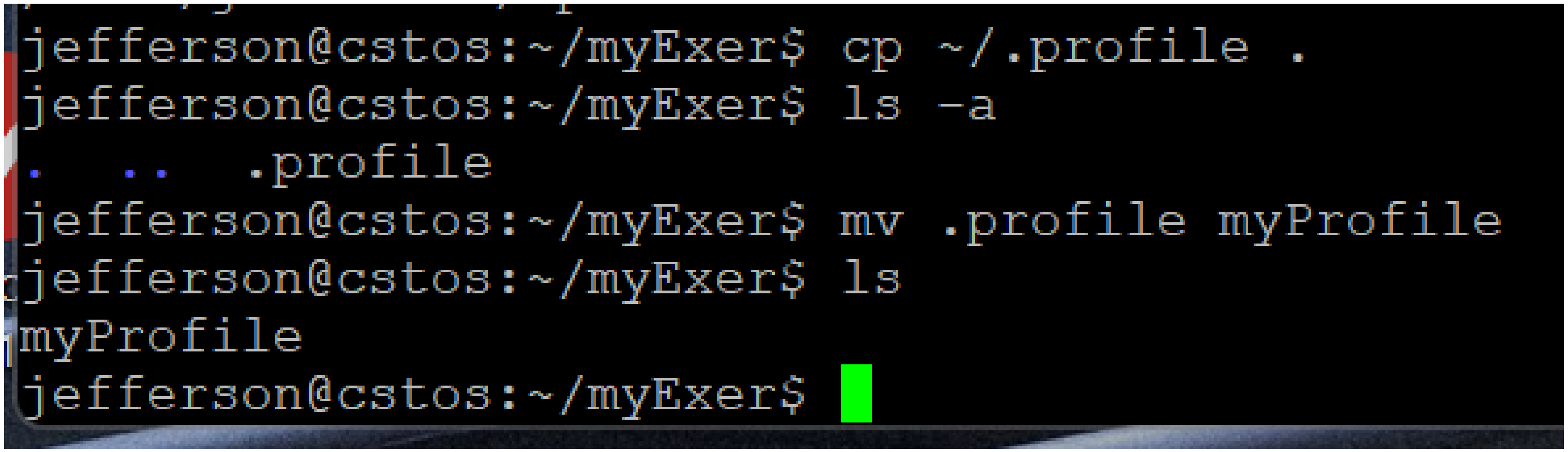
```
mv filename ..
```

Rename a filename

```
mv oldname newname
```

Task 2

- In the directory myExer, rename .profile to myProfile.



```
jefferson@cstos:~/myExer$ cp ~/.profile .
jefferson@cstos:~/myExer$ ls -a
.   ..   .profile
jefferson@cstos:~/myExer$ mv .profile myProfile
jefferson@cstos:~/myExer$ ls
myProfile
jefferson@cstos:~/myExer$ 
```

Task 2

- See the content of the file myProfile:
  - **cat myProfile**
  - **more myProfile**
  - **head myProfile**

  Etc.
- See next slide for explanations.

- When you become an expert Linux user, you can change your profile.
- For now, just look.

Displaying a file

Various ways to display a file in Unix

- **cat** : shows the contents of a file, all at once
  - Appropriate only for small file
- **more** : shows the contents of a file, screen by screen
  - Hit spacebar for next screen, q to quit.
- **less** : also shows the contents of a file, screen by screen
- **head** : show the specified number of lines from top of a file
- **tail** : show the specified number of lines from bottom of a file

Linux Command rm

- Remove files and directories:
  - **rm [option] *item***

  where item is one or more files or directories

Option	Long option	Meaning
-i	--interative	Before deleting an existing file, prompt the user for confirmation. **If this option is not specified, rm will silently delete files.**
-r	--recursive	Recursively delete directories. This means that if a directory being deleted has subdirectories, delete them too. To delete a directory, this option must be specified.
-f	--force	Ignore nonexistent files and do not prompt. This overrides the --interactive option.
-v	--verbose	Display informative messages as the deletion is performed.

File Command: rm

- Remove files “recursively”: **rm –r**
  - Use for removing **all files and sub directories**
- Be very careful, deletions are permanent in Unix/Linux
  - Make sure your command is correct before doing this.

- Section 2 – Test 2 covers up to here.

Remarks

- The following slides are **optional**.
  - They are very important; learn them yourself.
- We gave you a very brief intro on Linux and a glimpse of what it can do.
- Hopefully in future classes (e.g. OS, etc.) you will learn more.
- There is nothing to submit for the tasks, but this week's material (other than the optional part) is covered in Test 2.

File permissions

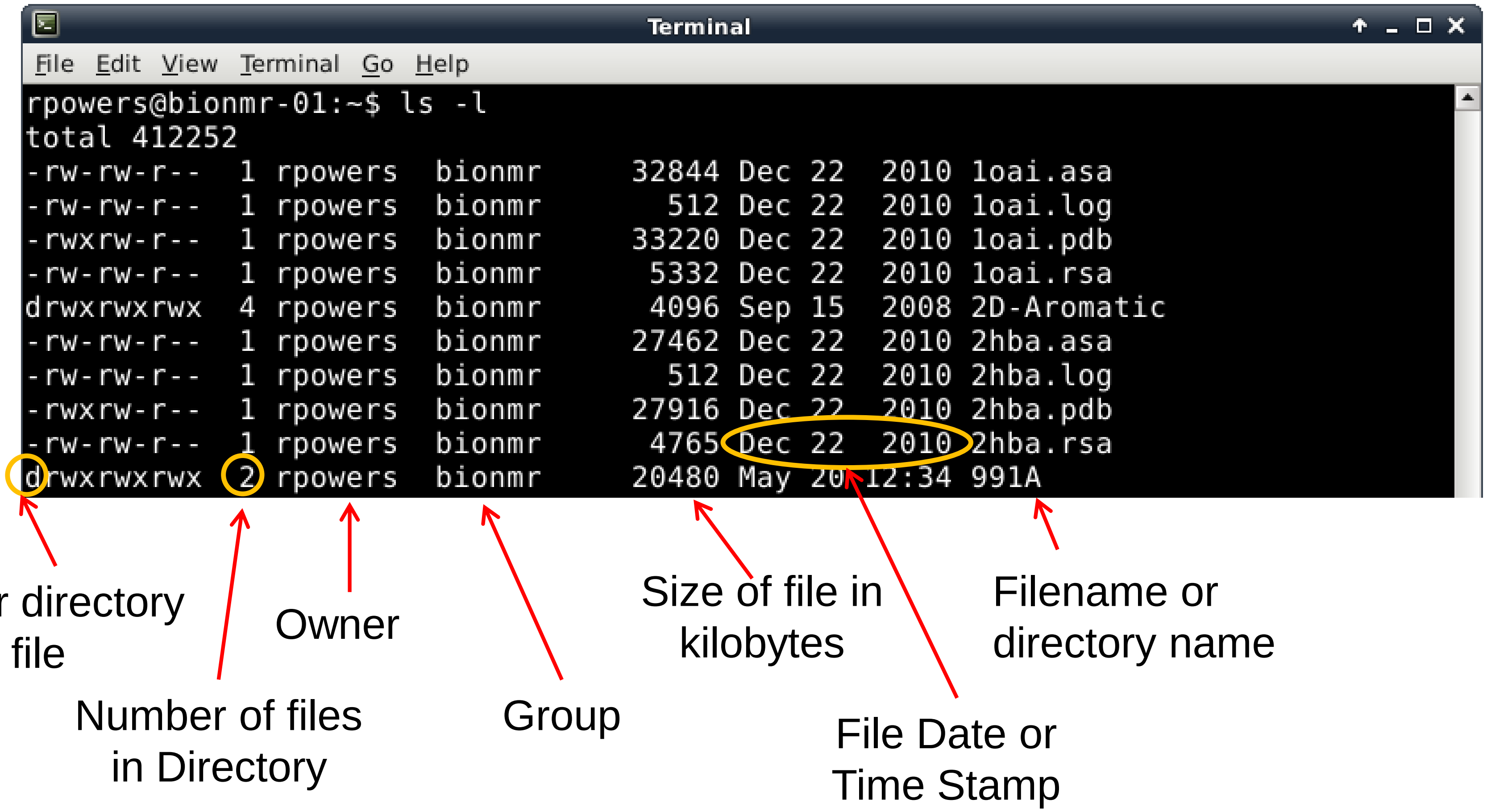
- Each file in Unix/Linux has an associated permission level
- There are 3 types of users for files or directories.
  - Owner: generally you, the person who created the file or directory.
  - Group: team members whom the owner wishes to share the files.
  - Public: Other users who has access to the server.
- Use "**ls -l *filename***" to see permission level of a file

Permission levels

- "r" means permission to "read" the file.
- "w" means permission to "write" (i.e. edit) the file.
- "x" means permission to "execute"
  - For a file, "x" means permission to run the code.
  - For a directory, "x" means permission to list the content of the directory.
- The owner of the file or directory has the authority to set the permission levels for that file or directory.

Permissions

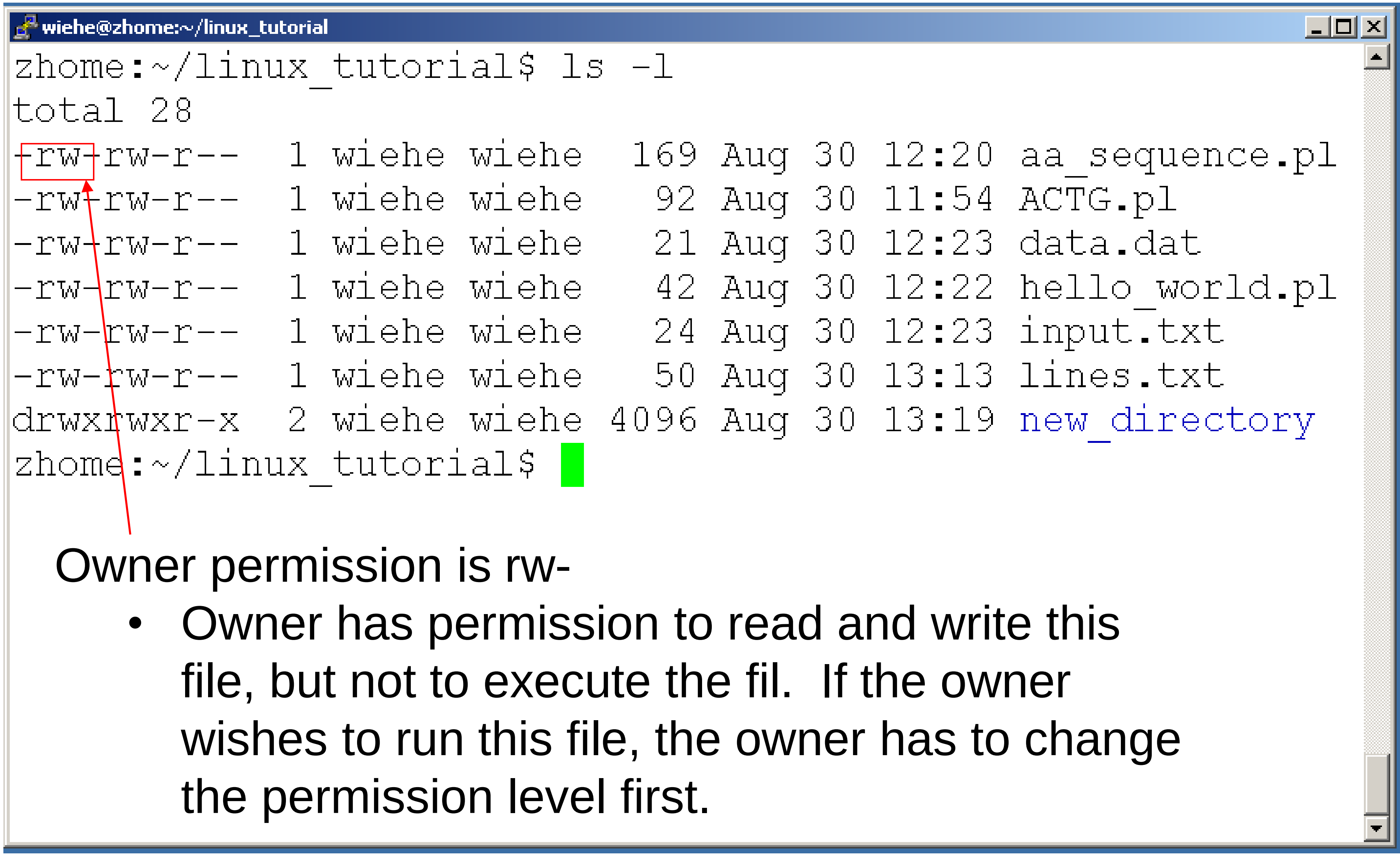
- You can't read, write, edit or execute a file without permission!



```
rpowers@bionmr-01:~$ ls -l
total 412252
-rw-rw-r--  1 rpowers  bionmr     32844 Dec 22   2010 1oai.asa
-rw-rw-r--  1 rpowers  bionmr       512 Dec 22   2010 1oai.log
-rwxrw-r--  1 rpowers  bionmr     33220 Dec 22   2010 1oai.pdb
-rw-rw-r--  1 rpowers  bionmr      5332 Dec 22   2010 1oai.rsa
drwxrwxrwx  4 rpowers  bionmr      4096 Sep 15   2008 2D-Aromatic
-rw-rw-r--  1 rpowers  bionmr     27462 Dec 22   2010 2hba.asa
-rw-rw-r--  1 rpowers  bionmr       512 Dec 22   2010 2hba.log
-rwxrw-r--  1 rpowers  bionmr     27916 Dec 22   2010 2hba.pdb
-rw-rw-r--  1 rpowers  bionmr      4765 Dec 22   2010 2hba.rsa
drwxrwxrwx  2 rpowers  bionmr     20480 May  20 12:34 991A
```

d for directory
- for file
Owner
Size of file in kilobytes
Filename or directory name
Number of files in Directory
Group
File Date or Time Stamp

File Permissions

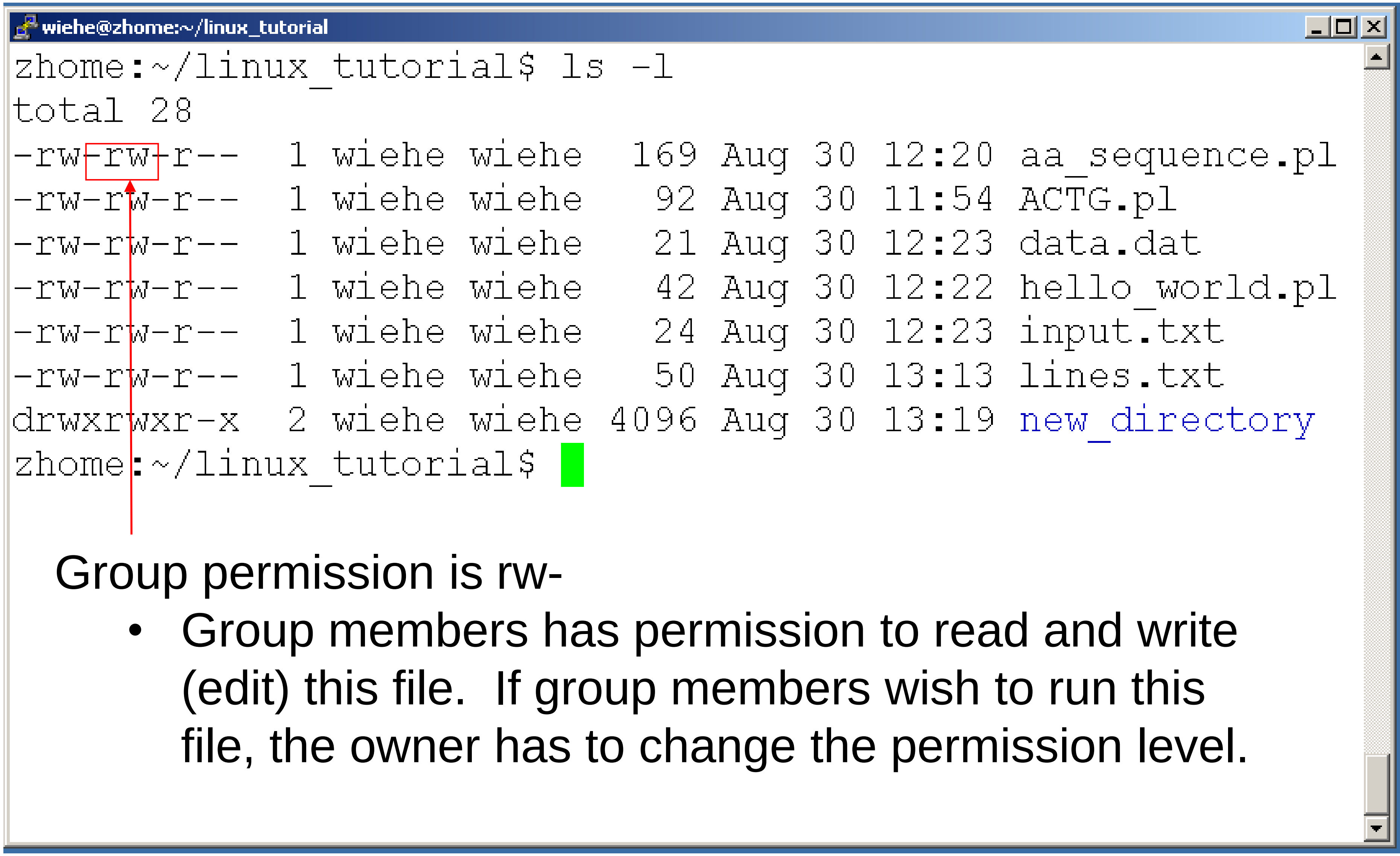


```
zhome:~/linux_tutorial$ ls -l
total 28
-rw-rw-r--   1 wiehe wiehe  169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--   1 wiehe wiehe   92 Aug 30 11:54 ACTG.pl
-rw-rw-r--   1 wiehe wiehe   21 Aug 30 12:23 data.dat
-rw-rw-r--   1 wiehe wiehe   42 Aug 30 12:22 hello_world.pl
-rw-rw-r--   1 wiehe wiehe   24 Aug 30 12:23 input.txt
-rw-rw-r--   1 wiehe wiehe   50 Aug 30 13:13 lines.txt
drwxrwxr-x   2 wiehe wiehe 4096 Aug 30 13:19 new_directory
zhome:~/linux_tutorial$
```

Owner permission is rw-

- Owner has permission to read and write this file, but not to execute the fil. If the owner wishes to run this file, the owner has to change the permission level first.

File Permissions

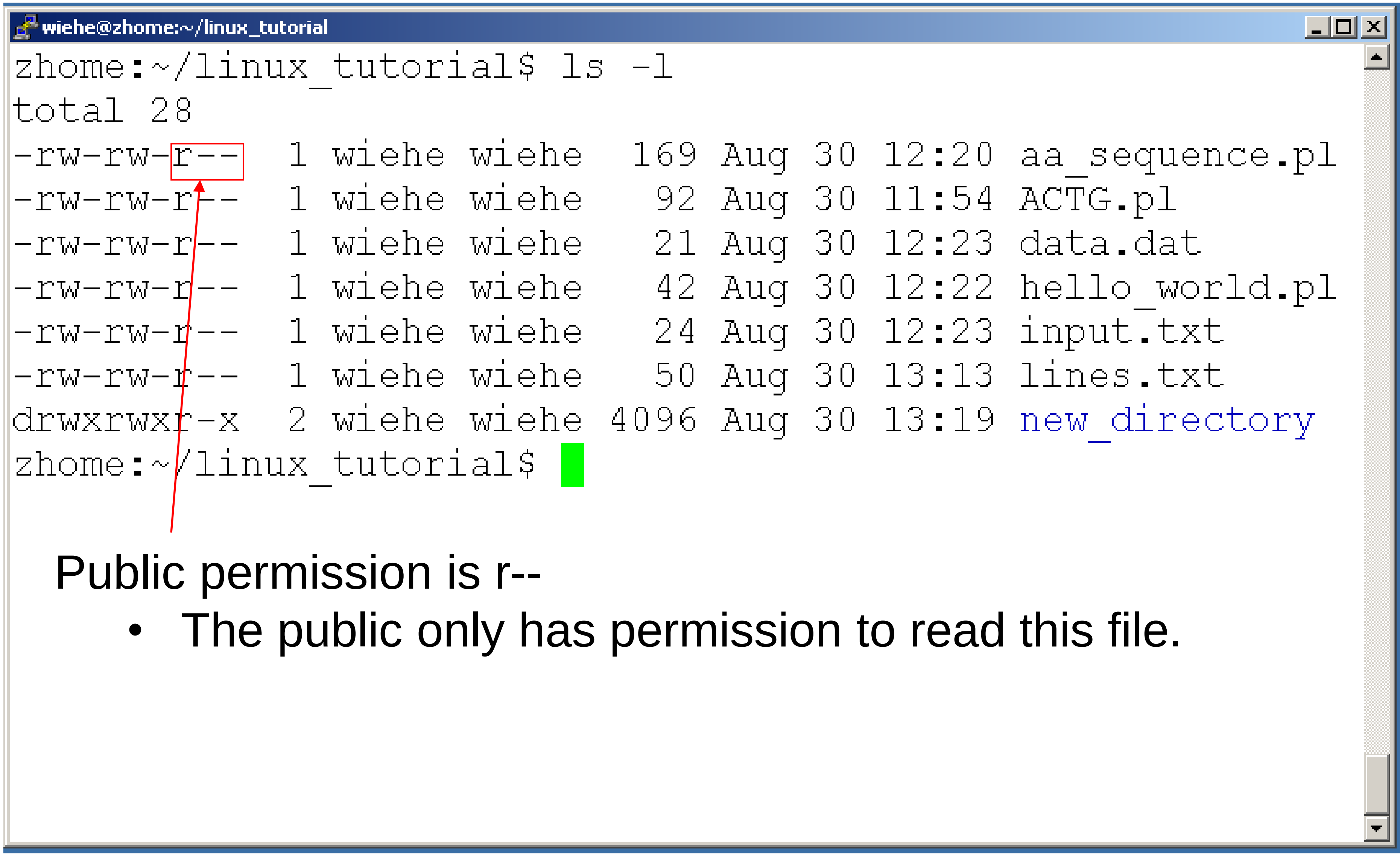


```
zhome:~/linux_tutorial$ ls -l
total 28
-rw-rw-r--   1 wiehe wiehe  169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--   1 wiehe wiehe   92 Aug 30 11:54 ACTG.pl
-rw-rw-r--   1 wiehe wiehe   21 Aug 30 12:23 data.dat
-rw-rw-r--   1 wiehe wiehe   42 Aug 30 12:22 hello_world.pl
-rw-rw-r--   1 wiehe wiehe   24 Aug 30 12:23 input.txt
-rw-rw-r--   1 wiehe wiehe   50 Aug 30 13:13 lines.txt
drwxrwxr-x   2 wiehe wiehe 4096 Aug 30 13:19 new_directory
zhome:~/linux_tutorial$
```

Group permission is rw-

- Group members has permission to read and write (edit) this file. If group members wish to run this file, the owner has to change the permission level.

File Permissions



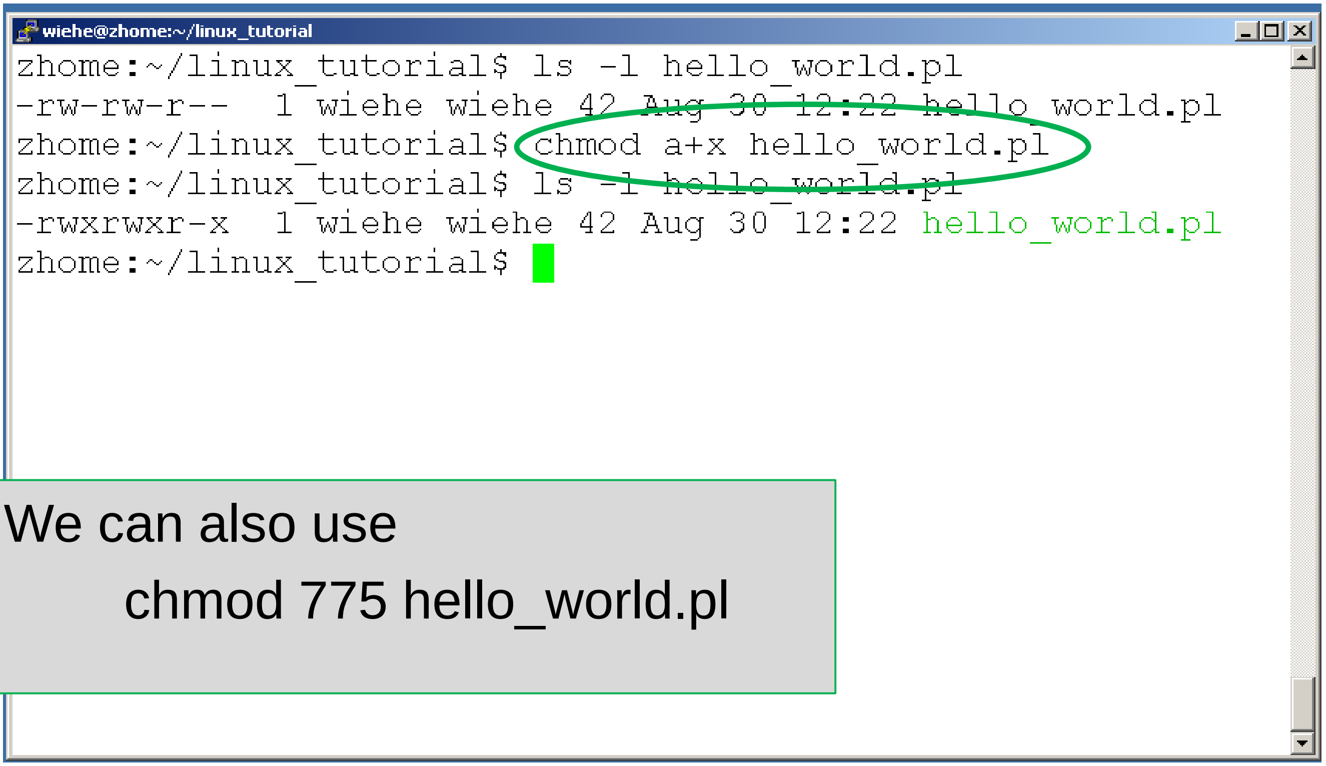
```
zhome:~/linux_tutorial$ ls -l
total 28
-rw-rw-r--  1 wiehe wiehe  169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--  1 wiehe wiehe   92 Aug 30 11:54 ACTG.pl
-rw-rw-r--  1 wiehe wiehe   21 Aug 30 12:23 data.dat
-rw-rw-r--  1 wiehe wiehe   42 Aug 30 12:22 hello_world.pl
-rw-rw-r--  1 wiehe wiehe   24 Aug 30 12:23 input.txt
-rw-rw-r--  1 wiehe wiehe   50 Aug 30 13:13 lines.txt
drwxrwxr-x  2 wiehe wiehe 4096 Aug 30 13:19 new_directory
zhome:~/linux_tutorial$
```

Public permission is r--

- The public only has permission to read this file.

Changing the permission level: chmod

- If you own the file, you can change it's permissions with "chmod"
  - Syntax: chmod [**u**ser/**g**roup/**o**thers/**a**ll]+[permission] [file(s)]
  - Below we grant execute permission to all:



```
zhome:~/linux_tutorial$ ls -l hello_world.pl
-rw-rw-r--  1 wiehe wiehe 42 Aug 30 12:22 hello_world.pl
zhome:~/linux_tutorial$ chmod a+x hello_world.pl
zhome:~/linux_tutorial$ ls -l hello_world.pl
-rwxrwxr-x  1 wiehe wiehe 42 Aug 30 12:22 hello_world.pl
zhome:~/linux_tutorial$ 
```

- We can also use

  chmod 775 hello_world.pl

Permissions

Meaning of chmod 775 filename

Think of the permission settings as a series of bits :

```
rwx rwx rwx = 111 111 111
rw- rw- rw- = 110 110 110
rwx --- --- = 111 000 000
```

and so on...

```
rwx = 111 in binary = 7
rw- = 110 in binary = 6
r-x = 101 in binary = 5
r-- = 100 in binary = 4
-xx = 011 in binary = 3
-x- = 010 in binary = 2
--x = 001 in binary = 1
--- = 000 in binary = 0
```

So for chmod 775,
owner has permission 7 or rwx,
group has permission 7 or rwx,
public has permission 5 or r-x.

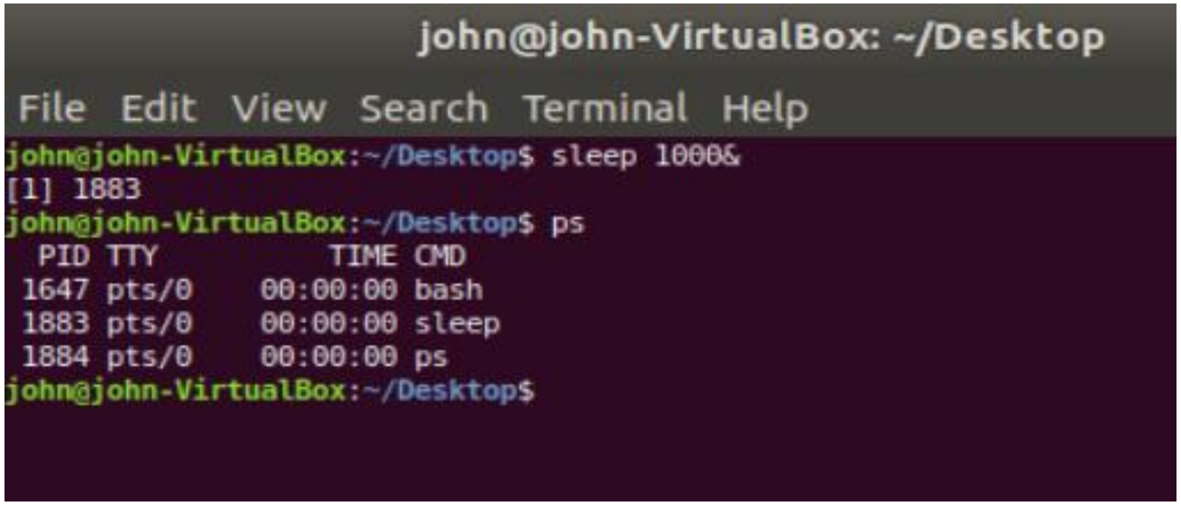
Running a program (a.k.a. a job)

- Make sure the program has executable permissions
- Use “./” to run the program
  - E.g. **./hello_world.pl**
  - “.” means local directory.

Command: ps

- To view the processes that you're running:

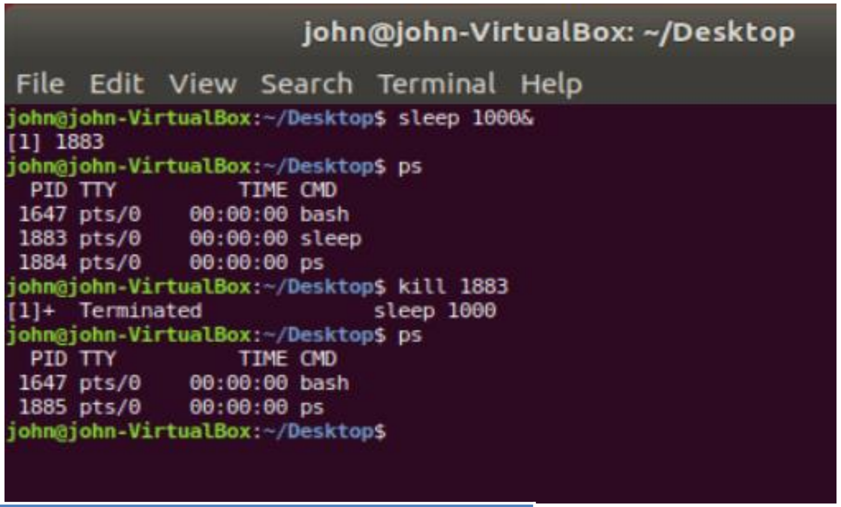
ps	Lists processes that you own
ps –l	Generates a long listing of your processes
ps –f	Outputs a full listing of processes that you own
ps –u login	Lists processes that are owned by the user whose login ID is login
ps –t nn	Lists processes that are associated with the workstation tty
ps –ef	Prints information about all processes
ps –aux	Prints information about all processes


john@john-VirtualBox: ~/Desktop
File Edit View Search Terminal Help

```
john@john-VirtualBox:~/Desktop$ sleep 1000&
[1] 1883
john@john-VirtualBox:~/Desktop$ ps
  PID TTY          TIME CMD
 1647 pts/0    00:00:00 bash
 1883 pts/0    00:00:00 sleep
 1884 pts/0    00:00:00 ps
john@john-VirtualBox:~/Desktop$
```

Command: kill

- To terminate a process use “kill”
  $kill pid

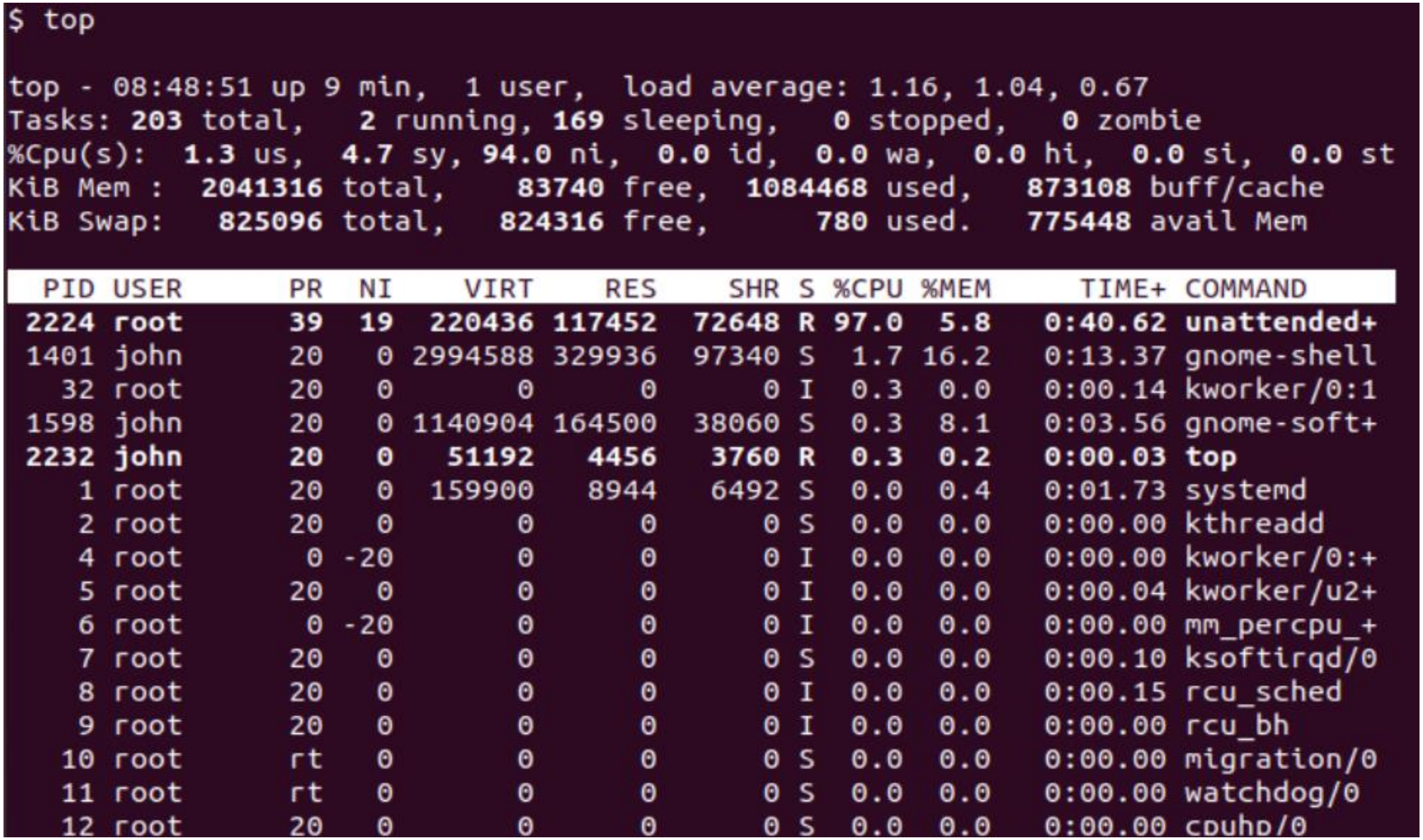

john@john-VirtualBox: ~/Desktop
File Edit View Search Terminal Help

```
john@john-VirtualBox:~/Desktop$ sleep 1000&
[1] 1883
john@john-VirtualBox:~/Desktop$ ps
  PID TTY          TIME CMD
 1647 pts/0    00:00:00 bash
 1883 pts/0    00:00:00 sleep
 1884 pts/0    00:00:00 ps
john@john-VirtualBox:~/Desktop$ kill 1883
[1]+  Terminated              sleep 1000
john@john-VirtualBox:~/Desktop$ ps
  PID TTY          TIME CMD
 1647 pts/0    00:00:00 bash
 1885 pts/0    00:00:00 ps
john@john-VirtualBox:~/Desktop$
```

kill PID	Terminates a process by sending a software terminate signal
kill -1 PID	Hangs up communication links to a process
kill -2 PID	Ends a process by sending a process interrupt signal
kill -3 PID	Brings a process to a conclusion by issuing a process quit signal
kill -6 PID	Instructs a process to end by issuing the Abort Process signal
kill -15 PID	Instructs a process to end by issuing the software termination signal
kill -9 PID	Kills a process.

Command: top

- To view the CPU usage of all processes:



```
$ top

top - 08:48:51 up 9 min,  1 user,  load average: 1.16, 1.04, 0.67
Tasks: 203 total,   2 running, 169 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.3 us,  4.7 sy, 94.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  2041316 total,    83740 free,  1084468 used,   873108 buff/cache
KiB Swap:   825096 total,   824316 free,      780 used.   775448 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 2224 root      39  19  220436 117452  72648 R  97.0  5.8   0:40.62 unattended+
 1401 john      20   0 2994588 329936  97340 S   1.7 16.2   0:13.37 gnome-shell
   32 root      20   0       0      0      0 I   0.3  0.0   0:00.14 kworker/0:1
 1598 john      20   0 1140904 164500  38060 S   0.3  8.1   0:03.56 gnome-soft+
 2232 john      20   0   51192   4456   3760 R   0.3  0.2   0:00.03 top
    1 root      20   0  159900   8944   6492 S   0.0  0.4   0:01.73 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:+
    5 root      20   0       0      0      0 I   0.0  0.0   0:00.04 kworker/u2+
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_+
    7 root      20   0       0      0      0 S   0.0  0.0   0:00.10 ksoftirqd/0
    8 root      20   0       0      0      0 I   0.0  0.0   0:00.15 rcu_sched
    9 root      20   0       0      0      0 I   0.0  0.0   0:00.00 rcu_bh
   10 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/0
   12 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/0
```

Task 3

- See what happens if you
  - Change the permission of the file myProfile.
    - Use   chmod 664; what are the permission levels?
  - Run process (ps) command
    - Don’t kill any process until you know what you are doing.
  - Check out the process using top CPU in your computer.

Input/Output Redirection(“piping”)

| (pipe) - passes output of one Linux command to the input of a second command

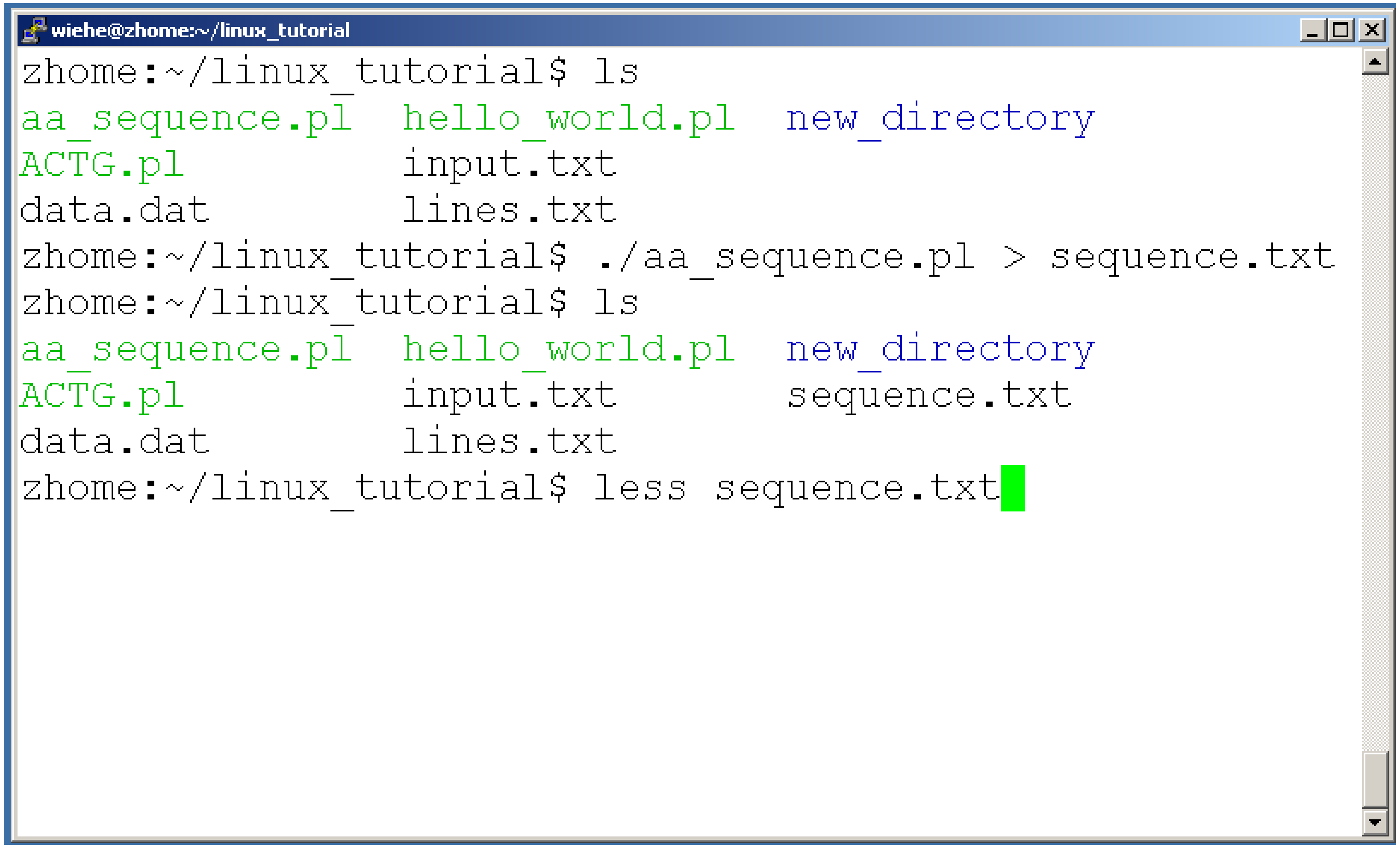
- **Example: ls |wc (wc** – counts the number of characters, words and lines)
- Not limited to just one pipe, can string multiple pipes together

>, < - redirection of files

- **command >** *filename* – output of command (or program) is sent to a file called *filename* instead of being displayed on the screen
  - **Example: ls >** ***file_list***
- **command <** *filename* – the file *filename* is the input to the command or program
  - **Example: xplor** < *psf.inp*

command > filename	**Send the output of the command to the filename**
command >> filename	Append the output of the command to the filename
command1 | command2	Make the output of command1 the input of command2

A few examples of piping

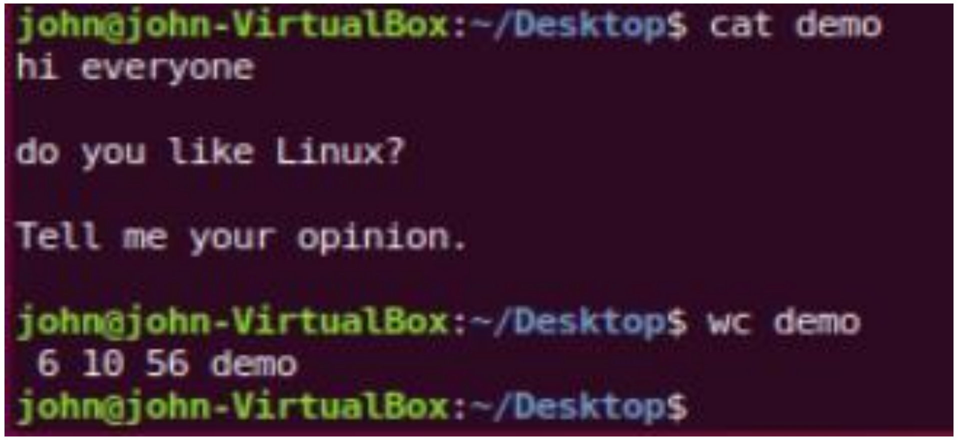


wiehe@zhome:~/linux_tutorial

```
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl         input.txt
data.dat        lines.txt
zhome:~/linux_tutorial$ ./aa_sequence.pl > sequence.txt
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl         input.txt       sequence.txt
data.dat        lines.txt
zhome:~/linux_tutorial$ less sequence.txt
```

Command: wc

- Word Count: count the characters, words, and lines
- The first column in the output is lines, the second is words, and the last is characters

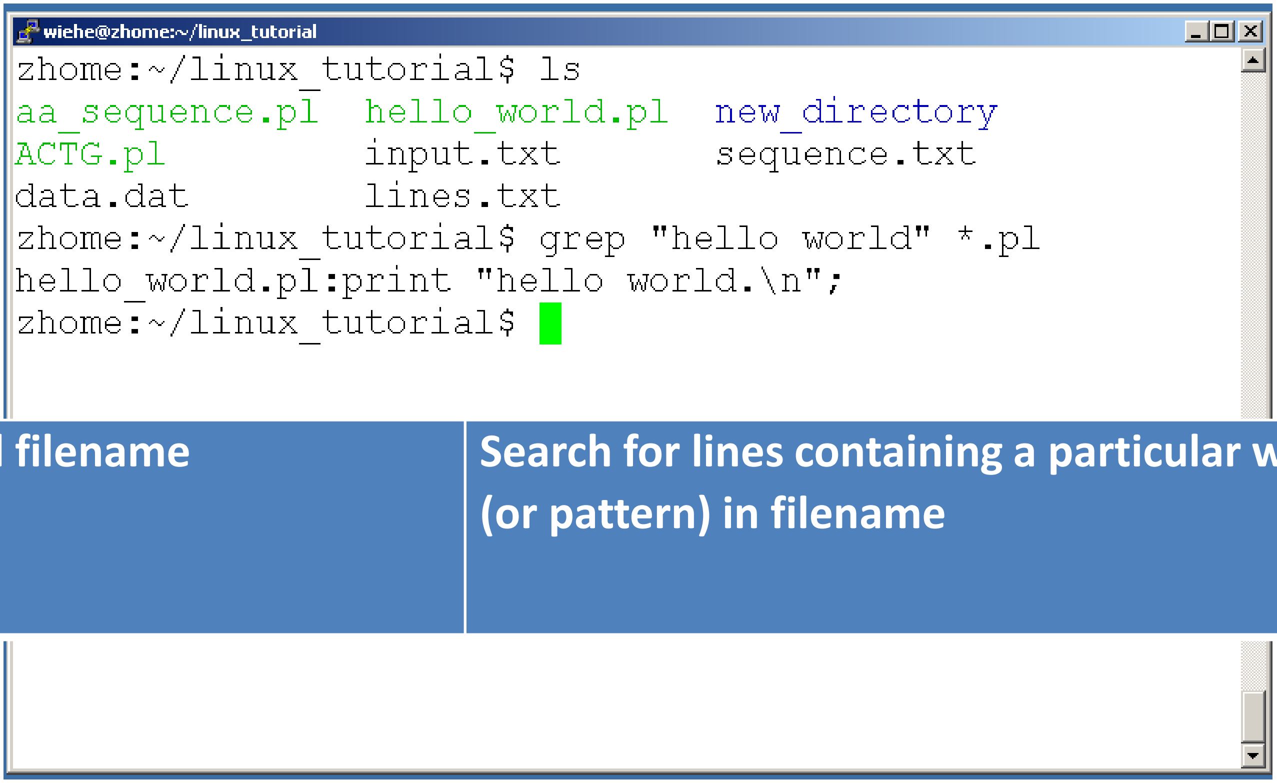


```
john@john-VirtualBox:~/Desktop$ cat demo
hi everyone

do you like Linux?

Tell me your opinion.

john@john-VirtualBox:~/Desktop$ wc demo
 6 10 56 demo
john@john-VirtualBox:~/Desktop$
```

wc filename	**Count the lines, words, characters in filename**
wc –l filename	Count the lines in filename, including empty lines
wc –c filename	Count the characters in filename
wc –w filename	Count the words in filename

Command: grep

- To search files in a directory for a specific string use "grep"



```
zhome:~/linux_tutorial$ ls
aa_sequence.pl   hello_world.pl   new_directory
ACTG.pl          input.txt        sequence.txt
data.dat         lines.txt
zhome:~/linux_tutorial$ grep "hello world" *.pl
hello_world.pl:print "hello world.\n";
zhome:~/linux_tutorial$
```

grep word filename	**Search for lines containing a particular word (or pattern) in filename**

Command: diff

- To compare to files for differences use “diff”
  - Try: **diff /dev/null hello.txt**
  - /dev/null is a special address -- it is always empty, and anything moved there is deleted

diff file1 file2	**Show lines that are different in each file and how to modify the first file to match the second file.**

Command: **comm**

- To compare two files use “comm”

comm file1 file2	**Compare file1 and file2 and show the lines common and unique in each of two files**

Background Computations

- For long computations, don't want the process directly associated with the window or shell
  - Window must remain open and active during computation
  - Window is "locked" until the program is finished
  - Computations will be stopped if the window is closed
  - A intense calculation can overwhelm the shell environment, leading to the window crashing or even slow down your computer
  - Output displays on window can be lost, lock window or crash computer

- Instead, submit your "job" to the "background"
  - Lowers the calculations priority to access the CPU
  - Any interactive calculation has the highest priority
  - **Example:** background - xplor < psf.inp > psf.out **&**

    interactive - xplor < psf.inp

- Use **ps** command to monitor status of background jobs

Linux Command Summary

Network: ssh, scp, ping, telnet, nslookup, wget

Shells: BASH, TCSH, alias, watch, clear, history, chsh, echo, set, setenv, xargs

System Information: w, whoami, man, info, which, free, echo, date, cal, df

Command Information: man, info

Symbols: |, >, >>, <, &, >&, 2>&1, ;, ~, ., .., $!, !:<n>, !<n>

Filters: grep, egrep, more, less, head, tail

Hotkeys: <ctrl><c>, <ctrl><d>, <ctrl><z>

File System: ls, mkdir, cd, pwd, mv, ln, touch, cat, file, find, diff, cmp, /net/<hostname>/<path>, mount, du, df, chmod

Line Editors: awk, sed

File Editors: vi, vim, gvim, emacs

Process Management: ps, top, kill, killall, fg, bg

For a Complete List of Linux Commands and Explanations see

http://linuxcommand.org/

Or the book “Linux in a Nutshell”

Unix Web Resources

- http://www.ee.surrey.ac.uk/Teaching/Unix/
- http://www.ugu.com/sui/ugu/show?help.beginners
- http://en.wikipedia.org/wiki/Unix