

2014-04-25.sagews

April 25, 2014

Contents

1	Math 480b Sage Course	1
1.1	The Command Line Terminal	1
1.2	April 25, 2014	1
1.3	The Command Line	1
1.4	History	2
1.5	History	2
1.6	What makes the terminal so powerful	5
1.7	Summary of key ideas	5

1 Math 480b Sage Course

1.1 The Command Line Terminal

1.2 April 25, 2014

Screencast: REMEMBER!!!!!! (with sound)

Plan

- Homework your grading of hw3 is due today at 6pm. I will collect and redistribute it by Saturday morning so at least you'll get some feedback on your project. your hw4 is due Sunday at 6pm.
- Lecture today on Command Line Terminal

1.3 The Command Line

- We will talk about bash (=bourne again shell), which is by far the most popular.
- This is what you get when you click +New;Command Line Terminal in SMC. Its also what you get when you open a terminal on Linux or OS X. It is NOT what you get when you open cmd.exe on Windows.
- You can execute all of the (non-interactive) commands we will mention here in a Sage worksheet or from the sage command-line terminal or from an IPython notebook by preceeding them with !

```
!pwd  
/projects/74af30b7-ad25-4308-a02e-c71fcd84de6e/sage2014/lectures/2014-04-25
```

```
!ls
2014-04-25.sagews
```

- You type stuff at a prompt and output appears. The command line terminal is extremely powerful. It mostly solves a similar problem to OS X's finder and other file explorers, but in a completely different way. Some things that are very hard in a graphical UI become utterly trivial in the command line, and conversely.
- The most important basic commands: memorize these
 - pwd = print working directory (where you are like the folder you are browsing)
 - ls = list the files in the working directory
 - cd path = change directory; use forward slashes. Use .. to go up.
 - mv src1 src2 ... dest = move a file (or files) or directory from one place to another (in particular, rename)
 - cp src1 src2 ... dest = copy a file from one place to another; use cp -r to recursively copy a directory
 - man command = manual about how to use a command
- Illustrate each of these in a terminal. (NOTE: In SMC, the terminal history currently vanishes after a while this will likely change though.)

1.4 History

- get it with the up and down arrows.
- type history to see it.
- type ![number] to re-run the numbered command.

1.5 History

```
# These guys created Unix in the 70s...
salvus.file("Ken_n_dennis.jpg")
```



```
# Actual 1979 Unix (Version 7) running:  
salvus.file("unix1979.png")
```

```

-rwxr-xr-x 1 sys      52850 Jun  8  1979 hptmunix
drwxrwxr-x 2 bin       320 Sep 22 05:33 lib
drwxrwxr-x 2 root      96 Sep 22 05:46 mdec
-rwxr-xr-x 1 root    50990 Jun  8  1979 rkunix
-rwxr-xr-x 1 root    51982 Jun  8  1979 rl2unix
-rwxr-xr-x 1 sys     51790 Jun  8  1979 rphtunix
-rwxr-xr-x 1 sys     51274 Jun  8  1979 rptmunix
drwxrwxrwx 2 root      48 Sep 22 05:50 tmp
drwxrwxr-x12 root     192 Sep 22 05:48 usr
# ls -l /usr
total 11
drwxrwxr-x 3 bin      128 Sep 22 05:45 dict
drwxrwxrwx 2 dmr       32 Sep 22 05:48 dmr
drwxrwxr-x 5 bin     416 Sep 22 05:46 games
drwxrwxr-x 3 sys     496 Sep 22 05:42 include
drwxrwxr-x10 bin     528 Sep 22 05:43 lib
drwxrwxr-x11 bin     176 Sep 22 05:45 man
drwxrwxr-x 3 bin     208 Sep 22 05:46 mdec
drwxrwxr-x 2 bin      80 Sep 22 05:46 pub
drwxrwxr-x 6 root      96 Sep 22 05:45 spool
drwxrwxr-x13 root    208 Sep 22 05:42 src
# ls -l /usr/dmr
total 0
# █

```

It looks similar 35 years later!

```

!ls -l /usr
total 184
drwxr-xr-x  2 root root 69632 Apr 25 13:56 bin
drwxr-xr-x  2 root root  4096 Mar  2 02:21 games
drwxr-xr-x 107 root root 20480 Mar 25 05:59 include
drwxr-xr-x 159 root root 45056 Apr 25 13:56 lib
drwxr-xr-x  3 root root  4096 Jan 16 03:43 lib32
drwxr-xr-x  3 root root  4096 Apr 25 2013 libexec
drwxr-xr-x 13 root root  4096 Nov 27 05:54 local
drwxr-xr-x  2 root root 12288 Apr  9 22:20 sbin
drwxr-xr-x 237 root root 12288 Mar 25 05:59 share
drwxr-xr-x 12 root root  4096 Mar 20 03:04 src

```

1.6 What makes the terminal so powerful

- Use tab completion (in the actual terminal) to complete file names
- You can use patterns to specify the files or directories that are arguments to commands
- You can redirect the input or output of a command.
- You can combine commands together via pipes (sort of like composing functions).
- You can temporarily pause (control-Z) and restart (fg) commands (and much more).
- There are thousands (!) of additional commands like the above, which all work in a uniform way.

Slowly and carefully illustrate the above points using the following commands:

ls, grep, du, man, sage, ipython, gp, git

1.7 Summary of key ideas

- The man command documents every command
- Use patterns to specify filenames
- Use `foo [...] — bar [...]` to make the output of foo be the input to bar
- Use `foo [...] output_file` to redirect the output of foo to the given file, and Use `foo [...] input_file` to make input to foo come from the given file.
- The most important commands are: man, ls, cd, mv, cp

Other: In SMC (and OS X), you can open a file or folder in the graphical interface by typing `open filename`