GRUNDLAGEN KOMMANDOZEILE
CRASH COURSE

KSBG 2016 | EF Informatik

Name und Vorname

Klasse

Quellenangaben:

- Shaw, Zed A.: Learn Ruby the Hard Way, Third Edition, Addison-Wesley, 2015
- Helpfiles für Powershell Windows
- Helpfiles für Linux Terminal

Inhaltsverzeichnis

1 Introduction

This appendix is a crash course in using the command line to make your computer perform tasks. As a crash course, it's not as detailed or extensive as my other books. It is simply designed to get you barely capable enough to start using your computer like a real programmer does. When you're done with this appendix, you will be able to give most of the basic commands that every shell user touches every day. You'll understand the basics of directories and a few other concepts.

2 The Setup

In this appendix you will be instructed to do three things:

1. Do some things in your shell (command line, Terminal, PowerShell).
2. Learn about what you just did.
3. Do more on your own.

For this first exercise you'll be expected to get your Terminal open and working so that you can do the rest of the appendix.

2.1 Do This

Get your Terminal, shell, or PowerShell working so you can access it quickly and know that it works.

2.1.1 Mac OSX

For Mac OSX you'll need to do this:
Hold down the command key and hit the spacebar.
In the top right the blue "search bar" will pop up.
Type: terminal
Click on the Terminal application that looks kind of like a black box.
This will open Terminal.
You can now go to your dock and CTRL-click to pull up the menu, then select Options->Keep In dock.
Now you have your Terminal open and it's in your dock so you can get to it.

2.1.2 Linux

I'm assuming that if you have Linux then you already know how to get at your Terminal. Look through the menu for your window manager for anything named "Shell" or "Terminal."

2.1.3 Windows

On Windows we're going to use PowerShell. People used to work with a program called cmd.exe, but it's not nearly as usable as PowerShell. If you have Windows 7 or later, do this:
Click Start.
In "Search programs and files" type: powershell
Hit Enter.

If you don't have Windows 7, you should seriously consider upgrading. If you still insist on not upgrading then you can try installing it from Microsoft's download center. Search online to find "powershell downloads" for your version of Windows. You are on your own, though, since I don't have Windows XP, but hopefully the PowerShell experience is the same.

2.2 Overview Commands

2.2.1 Linux/Mac OSX

Take this list of commands and create index cards with the names on the left on one side, and the definitions on the other side. Drill them every day while continuing with the lessons in this appendix.

```
pwd                  print working directory
hostname             my computer's network name
mkdir                make directory
cd                   change directory
cd ~                 change directly to home directory
ls                   list directory
rmdir                remove directory
pushd                push directory
popd                 pop directory
cp                   copy a file or directory
mv                   move a file or directory
less                 page through a file
cat                  print the whole file
xargs                execute arguments
find                 find files
grep                 find things inside files
man                  read a manual page
apropos              find what man page is appropriate
env                  look at your environment
echo                 print some arguments
export               export/set a new environment variable
exit                 exit the shell
[command] --help     help for the command
[applicationName]    starts application out of the terminal
[applicationName] &  starts application in background mode
sudo                 DANGER! become super user root DANGER!
```

2.2.2 Windows

If you're using Windows then here's your list of commands:

```
pwd                  print working directory
hostname             my computer's network name
mkdir                make directory
cd                   change directory
ls                   list directory
rmdir                remove directory
pushd                push directory
popd                 pop directory
cp                   copy a file or directory
robocopy             robust copy
mv                   move a file or directory
more                 page through a file
type                 print the whole file
forfiles             run a command on lots of files
dir -r               find files
select-string        find things inside files
help                 read a manual page
helpctr              find what man page is appropriate
echo                 print some arguments
set                  export/set a new environment variable
exit                 exit the shell
runas                DANGER! become super user root DANGER!
```

Drill, drill, drill! Drill until you can say these phrases right away when you see that word. Then drill the inverse, so that you read the phrase and know what command will do that. You're building your vocabulary by doing this, but don't spend so much time you go nuts and get bored.

3 Speed-up tricks on Linux

Try to speed up your shell commands via the following hints :

- « tab » key : This feature can dramatically help you speed up typing commands. Just hit Tab while typing a command, option, or file name and the shell environment will automatically complete what you're typing or suggest options to you.

  Tab completion is especially useful when typing file names, directories, and paths. Rather than trying to type a long file name that may involve spaces and special characters you'll need to properly escape, you can just start typing the beginning of the name and press Tab.

  For example, if we have a long, complex file name beginning with the letter L, we'd just have to type L and press Tab to automatically complete it. If we had multiple file names beginning with L, we'd just need to type a bit more of the file's name before pressing Tab again.

- «up-arrow» key: Brings up the previous command I have entered. This is very useful, if I did a wrong entry before and I like to change it rapidly without typing everything from the beginning.

- --help : Many executable programs support a "--help" option that displays a description of the command's supported syntax and options. For example: cp --help lists a short description of the copy command with syntax and options.

- help : If you just type help followed by a command you will get a display with brief summaries of shell builtin commands.

4 Paths, Folders, Directories (pwd)

In this exercise you learn how to print your working directory with the pwd command.

4.1 Do This

I'm going to teach you how to read these "sessions" that I show you. You don't have to type everything I list here, just some of the parts:

- You do not type in the $ (Unix) or > (Windows). That's just me showing you my session so you can see what I got.
- You type in the stuff after $ or >, then hit Enter. So if I have $ pwd you type just pwd and hit Enter.
- You can then see what I have for output followed by another $ or > prompt. That content is the output and you should see the same output.

Let's do a simple first command so you can get the hang of this:

4.1.1 Linux/OSX

```
$ pwd
/Users/zedshaw
$
```

4.1.2 Windows

```
PS C:\Users\zed> pwd

Path
----
C:\Users\zed

PS C:\Users\zed>
```

4.2 You Learned This

Your prompt will look different from mine. You may have your user name before the $ and the name of your computer. On Windows it will probably look different too. The key is that you see the pattern of:

- There's a prompt.
- You type a command there. In this case, it's pwd.
- It printed something.

5 If You Get Lost

As you go through these instructions you may get lost. You may not know where you are or where a file is and have no idea how to continue. To solve this problem I am going to teach you the commands to type to stop being lost.

Whenever you get lost, it is most likely because you were typing commands and have no idea where you've ended up. What you should do is type pwd to print your current directory. This tells you where you are.

The next thing is you need to have a way of getting back to where you are safe, your home. To do this type cd ~ and you are back in your home.

This means if you get lost at any time type:

```
pwd
cd ~
```

The first command pwd tells you where you are. The second command cd ~ takes you home so you can try again.

5.1 Do This

Right now figure out where you are, and then go home using pwd and cd ~. This will make sure you are always in the right place.

6 Make A Directory (mkdir)

In this exercise you learn how to make a new directory (folder) using the mkdir command.

6.1 Do This

Remember! You need to go home first! Do your pwd then cd ~ before doing this exercise. Before you do all exercises in this appendix, always go home first!

6.1.1 Linux/OSX

```
$ pwd
$ cd ~
$ mkdir temp
$ mkdir temp/stuff
$ mkdir temp/stuff/things
$ mkdir -p temp/stuff/things/orange/apple/pear/grape
$
```

6.1.2 Windows

```
> pwd
> cd ~
> mkdir temp
    Directory: C:\Users\zed
Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:02 AM            temp

> mkdir temp/stuff
    Directory: C:\Users\zed\temp
Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:02 AM            stuff

> mkdir temp/stuff/things
    Directory: C:\Users\zed\temp\stuff

Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            things

> mkdir temp/stuff/things/orange/apple/pear/grape
    Directory: C:\Users\zed\temp\stuff\things\orange\apple\pear
Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            grape

>
```

This is the only time I'll list the pwd and cd ~ commands. They are expected in the exercises every time. Do them all the time.

6.2 You Learned This

Now we get into typing more than one command. These are all the different ways you can run mkdir. What does mkdir do? It make directories. Why are you asking that? You should be doing your index cards and getting your commands memorized. If you don't know that "mkdir makes directories" then keep working the index cards.

What does it mean to make a directory? You might call directories "folders." They're the same thing. All you did above is create directories inside directories inside of more directories. This is called a "path" and it's a way of saying "first temp, then stuff, then things and that's where I want it." It's a set of directions to the computer of where you want to put something in the tree of folders (directories) that make up your computer's hard disk.

7 Change Directory (cd)

In this exercise you learn how to change from one directory to another using the cd command.

7.1 Do This

I'm going to give you the instructions for these sessions one more time:

- You do not type in the $ (Unix) or > (Windows).
- You type in the stuff after this, then hit Enter. If I have $ cd temp you just type cd temp and hit Enter.
- The output comes after you hit Enter, followed by another $ or > prompt.
- Always go home first! Do pwd and then cd ~ so you go back to your starting point.

7.1.1 Linux/OSX

```
$ cd temp
$ pwd
~/temp
$ cd stuff
$ pwd
~/temp/stuff
$ cd things
$ pwd
~/temp/stuff/things
$ cd orange/
$ pwd
~/temp/stuff/things/orange
$ cd apple/
$ pwd
~/temp/stuff/things/orange/apple
$ cd pear/
$ pwd
~/temp/stuff/things/orange/apple/pear
$ cd grape/
$ pwd
~/temp/stuff/things/orange/apple/pear/grape
$ cd ..
$ cd ..
$ pwd
~/temp/stuff/things/orange/apple
$ cd ..
$ cd ..
$ pwd
~/temp/stuff/things
$ cd ../../..
$ pwd
~/
$ cd temp/stuff/things/orange/apple/pear/grape
$ pwd
~/temp/stuff/things/orange/apple/pear/grape
$ cd ../../../../../../../
$ pwd
~/
$
```

7.1.2 Windows

```
> cd temp
> pwd

Path
----
C:\Users\zed\temp
```

```
> cd stuff
> pwd

Path
----
C:\Users\zed\temp\stuff


> cd things
> pwd

Path
----
C:\Users\zed\temp\stuff\things


> cd orange
> pwd

Path
----
C:\Users\zed\temp\stuff\things\orange


> cd apple
> pwd

Path
----
C:\Users\zed\temp\stuff\things\orange\apple


> cd pear
> pwd

Path
----
C:\Users\zed\temp\stuff\things\orange\apple\pear


> cd grape
> pwd

Path
----
C:\Users\zed\temp\stuff\things\orange\apple\pear\grape


> cd ..
> cd ..
> cd ..
> pwd

Path
----
C:\Users\zed\temp\stuff\things\orange


> cd ../..
> pwd

Path
----
C:\Users\zed\temp\stuff


> cd ..
> cd ..
> cd temp/stuff/things/orange/apple/pear/grape
> cd ../../../../../../../
> pwd

Path
----
C:\Users\zed


>
```

7.2 You Learned This

You made all these directories in the last exercise, and now you're just moving around inside them with the cd command. In my session above I also use pwd to check where I am, so remember not to type the output that pwd prints. For example, on line 3 you see ~/temp but that's the output of pwd from the prompt above it. Do not type this in.

You should also see how I use the .. to move "up" in the tree and path.

8 List Directory (ls)

In this exercise you learn how to list the contents of a directory with the ls command.

8.1 Do This

Before you start, make sure you cd back to the directory above temp. If you have no idea where you are, use pwd to figure it out and then move there.

8.1.1 Linux/OSX

```
$ cd temp
$ ls
stuff
$ cd stuff
$ ls
things
$ cd things
$ ls
orange
$ cd orange
$ ls
apple
$ cd apple
$ ls
pear
$ cd pear
$ ls
$ cd grape
$ ls
$ cd ..
$ ls
grape
$ cd ../../../
$ ls
orange
$ cd ../../
$ ls
stuff
$
```

8.1.2 Windows

```
> cd temp
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            stuff


> cd stuff
> ls


    Directory: C:\Users\zed\temp\stuff


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            things
```

```
> cd things
> ls


    Directory: C:\Users\zed\temp\stuff\things


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            orange


> cd orange
> ls


    Directory: C:\Users\zed\temp\stuff\things\orange


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            apple


> cd apple
> ls


    Directory: C:\Users\zed\temp\stuff\things\orange\apple


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            pear


> cd pear
> ls


    Directory: C:\Users\zed\temp\stuff\things\orange\apple\pear


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            grape


> cd grape
> ls
> cd ..
> ls


    Directory: C:\Users\zed\temp\stuff\things\orange\apple\pear


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            grape


> cd ..
> ls


    Directory: C:\Users\zed\temp\stuff\things\orange\apple
```

```
Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            pear


> cd ../../..
> ls


    Directory: C:\Users\zed\temp\stuff


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            things


> cd ..
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            stuff


>
```

8.2 You Learned This

The ls command lists out the contents of the directory you are currently in. You can see me use cd to change into different directories and then list what's in them so I know which directory to go to next.

There are a lot of options for the ls command, but you'll learn how to get help on those later when we cover the help command.

9 Remove Directory (rmdir)

In this exercise you learn how to remove an empty directory.

9.1 Do This

9.1.1 Linux/OSX

```
$ cd temp
$ ls
stuff
$ cd stuff/things/orange/apple/pear/grape/
$ cd ..
$ rmdir grape
$ cd ..
$ rmdir pear
$ cd ..
$ ls
apple
$ rmdir apple
$ cd ..
$ ls
orange
$ rmdir orange
$ cd ..
$ ls
things
$ rmdir things
$ cd ..
$ ls
stuff
$ rmdir stuff
$ pwd
~/temp
$
```

WARNING: If you try to do rmdir on Mac OSX and it refuses to remove the directory even though you are positive it's empty, then there is actually a file in there called .DS_Store. In that case, type rm -rf <dir> instead (replace <dir> with the directory name).

9.1.2 Windows

```
> cd temp
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:03 AM            stuff


> cd stuff/things/orange/apple/pear/grape/
> cd ..
> rmdir grape
> cd ..
> rmdir pear
> cd ..
> rmdir apple
> cd ..
> rmdir orange
> cd ..
> ls
```

```
    Directory: C:\Users\zed\temp\stuff


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:14 AM            things


> rmdir things
> cd ..
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/17/2011   9:14 AM            stuff


> rmdir stuff
> pwd

Path
----
C:\Users\zed\temp


> cd ..
>
```

9.2 You Learned This

I'm now mixing up the commands so make sure you type them exactly and pay attention. Every time you make a mistake, it's because you aren't paying attention. If you find yourself making many mistakes, then take a break or just quit for the day. You've always got tomorrow to try again.

In this example you'll learn how to remove a directory. It's easy. You just go to the directory right above it, then type rmdir <dir>, replacing <dir> with the name of the directory to remove.

10 Moving Around (pushd, popd)

In this exercise you learn how to save your current location and go to a new location with pushd. You then learn how to return to the saved location with popd.

10.1 Do This

10.1.1 Linux/OSX

```
$ cd temp
$ mkdir -p i/like/icecream
$ pushd i/like/icecream
~/temp/i/like/icecream ~/temp
$ popd
~/temp
$ pwd
~/temp
$ pushd i/like
~/temp/i/like ~/temp
$ pwd
~/temp/i/like
$ pushd icecream
~/temp/i/like/icecream ~/temp/i/like ~/temp
$ pwd
~/temp/i/like/icecream
$ popd
~/temp/i/like ~/temp
$ pwd
~/temp/i/like
$ popd
~/temp
$ pushd i/like/icecream
~/temp/i/like/icecream ~/temp
$ pushd
~/temp ~/temp/i/like/icecream
$ pwd
~/temp
$ pushd
~/temp/i/like/icecream ~/temp
$ pwd
~/temp/i/like/icecream
$
```

10.1.2 Windows

```
> cd temp
> mkdir -p i/like/icecream


    Directory: C:\Users\zed\temp\i\like


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/20/2011  11:05 AM            icecream


> pushd i/like/icecream
> popd
> pwd

Path
----
C:\Users\zed\temp
```

```
> pushd i/like
> pwd

Path
----
C:\Users\zed\temp\i\like


> pushd icecream
> pwd

Path
----
C:\Users\zed\temp\i\like\icecream


> popd
> pwd

Path
----
C:\Users\zed\temp\i\like


> popd
>
```

10.2 You Learned This

You're getting into programmer territory with these commands, but they're so handy I have to teach them to you. These commands let you temporarily go to a different directory and then come back, easily switching between the two.

The pushd command takes your current directory and "pushes" it into a list for later, then it changes to another directory. It's like saying, "Save where I am, then go here."

The popd command takes the last directory you pushed and "pops" it off, taking you back there.

Finally, on Unix pushd, if you run it by itself with no arguments, will switch between your current directory and the last one you pushed. It's an easy way to switch between two directories. This does not work in PowerShell.

11 Making Empty Files (Touch, New-Item)

In this exercise you learn how to make an empty file using the touch (new-item on Windows) command.

11.1 Do This

11.1.1 Linux/OSX

```
$ cd temp
$ touch iamcool.txt
$ ls
iamcool.txt
$
```

11.1.2 Windows

```
> cd temp
> New-Item iamcool.txt -type file
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        12/17/2011   9:03 AM            iamcool.txt


>
```

11.2 You Learned This

You learned how to make an empty file. On Unix touch does this, and it also changes the times on the file. I rarely use it for anything other than making empty files. On Windows you don't have this command, so you learned how to use the New-Item command, which does the same thing but can also make new directories.

12 Copy a File (cp)

In this exercise you learn how to copy a file from one location to another with the cp command.

12.1 Do This

12.1.1 Linux/OSX

```
$ cd temp
$ cp iamcool.txt neat.txt
$ ls
iamcool.txt neat.txt
$ cp neat.txt awesome.txt
$ ls
awesome.txt iamcool.txt neat.txt
$ cp awesome.txt thefourthfile.txt
$ ls
awesome.txt  iamcool.txt  neat.txt  thefourthfile.txt
$ mkdir something
$ cp awesome.txt something/
$ ls
awesome.txt iamcool.txt  neat.txt  something  thefourthfile.txt
$ ls something/
awesome.txt
$ cp -r something newplace
$ ls newplace/
awesome.txt
$
```

12.1.2 Windows

```
> cd temp
> cp iamcool.txt neat.txt
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt


> cp neat.txt awesome.txt
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        12/22/2011   4:49 PM          0 awesome.txt
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt


> cp awesome.txt thefourthfile.txt
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
```

```
----                -------------     ------ ----
-a---        12/22/2011   4:49 PM          0 awesome.txt
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt


> mkdir something


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            something


> cp awesome.txt something/
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            something
-a---        12/22/2011   4:49 PM          0 awesome.txt
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt


> ls something


    Directory: C:\Users\zed\temp\something


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        12/22/2011   4:49 PM          0 awesome.txt


> cp -recurse something newplace
> ls newplace


    Directory: C:\Users\zed\temp\newplace


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        12/22/2011   4:49 PM          0 awesome.txt

>
```

12.2 You Learned This

Now you can copy files. It's simple to just take a file and copy it to a new one. In this exercise I also make a new directory and copy a file into that directory.

I'm going to tell you a secret about programmers and system administrators now. They are lazy. I'm lazy. My friends are lazy. That's why we use computers. We like to make computers do boring things for us. In the exercises so far you have been typing repetitive boring commands so that you can learn them, but usually it's not like this. Usually if you find yourself doing something boring and repetitive there's probably a programmer who has figured out how to make it easier. You just don't know about it.

The other thing about programmers is they aren't nearly as clever as you think. If you overthink what to type, then you'll probably get it wrong. Instead, try to imagine what the name of a command is to you and try it. Chances are that it's a name or some abbreviation similar to what you thought it was. If you still can't figure it out intuitively, then ask around and search online. Hopefully it's not something really stupid like ROBOCOPY.

13 Moving a File (mv)

In this exercise you learn how to move a file from one location to another using the mv command.

13.1 Do This

13.1.1 Linux/OSX

```
$ cd temp
$ mv awesome.txt uncool.txt
$ ls
newplace    uncool.txt
$ mv newplace oldplace
$ ls
oldplace    uncool.txt
$ mv oldplace newplace
$ ls
newplace   uncool.txt
$
```

13.1.2 Windows

```
> cd temp
> mv awesome.txt uncool.txt
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            newplace
d----        12/22/2011   4:52 PM            something
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt
-a---        12/22/2011   4:49 PM          0 uncool.txt


> mv newplace oldplace
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            oldplace
d----        12/22/2011   4:52 PM            something
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt
-a---        12/22/2011   4:49 PM          0 uncool.txt


> mv oldplace newplace
> ls newplace


    Directory: C:\Users\zed\temp\newplace


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        12/22/2011   4:49 PM          0 awesome.txt
```

```
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            newplace
d----        12/22/2011   4:52 PM            something
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt
-a---        12/22/2011   4:49 PM          0 uncool.txt


>
```

13.2 You Learned This

Moving files or, rather, renaming them. It's easy: give the old name and the new name.

14 View a File (less, MORE)

To do this exercise you're going to do some work using the commands you know so far. You'll also need a text editor that can make plain text (.txt) files. Here's what you do:

Open your text editor and type some stuff into a new file. On OSX this could be TextWrangler. On Windows this might be Notepad++. On Linux this could be gedit. Any editor will work.

Save that file to your desktop and name it test.txt.

In your shell use the commands you know to copy this file to your temp directory that you've been working with.

Once you've done that, complete this exercise.

14.1 Do This

14.1.1 Linux/OSX

```
$ less test.txt
[displays file here]
$
```

That"s it. To get out of less just type q (as in quit).

14.1.2 Windows

```
> more test.txt
[displays file here]
>
```

NOTE : In the above output I'm showing [displays file here] to "abbreviate" what that program shows. I'll do this when I mean to say, "Showing you the output of this program is too complex, so just insert what you see on your computer here and pretend I did show it to you." Your screen will not actually show this.

14.2 You Learned This

This is one way to look at the contents of a file. It's useful because, if the file has many lines, it will "page" so that only one screenful at a time is visible. In the Do More section you'll play with this some more.

15 Stream a File (cat)

You're going to do some more setup for this one so you get used to making files in one program and then accessing them from the command line. With the same text editor from the last exercise, create another file named test2.txt but this time save it directly to your temp directory.

15.1 Do This

15.1.1 Linux/OSX

```
$ less test2.txt
[displays file here]
$ cat test2.txt
I am a fun guy.
Don't you know why?
Because I make poems,
that make babies cry.
$ cat test.txt
Hi there this is cool.
$
```

15.1.2 Windows

```
> more test2.txt
[displays file here]
> cat test2.txt
I am a fun guy.
Don't you know why?
Because I make poems,
that make babies cry.
> cat test.txt
Hi there this is cool.
>
```

Remember that when I say [displays file here] I'm abbreviating the output of that command so I don't have to show you exactly everything.

15.2 You Learned This

Do you like my poem? Totally going to win a Nobel. Anyway, you already know the first command, and I'm just having you check that your file is there. Then you cat the file to the screen. This command just spews the whole file to the screen with no paging or stopping. To demonstrate that, I have you do this to the test.txt which should just spew a bunch of lines from that exercise.

16 Removing a File (rm)

In this exercise you learn how to remove (delete) a file using the rm command.

16.1 Do This

16.1.1 Linux

```
$ cd temp
$ ls
uncool.txt  iamcool.txt  neat.txt  something  thefourthfile.txt
$ rm uncool.txt
$ ls
iamcool.txt  neat.txt  something  thefourthfile.txt
$ rm iamcool.txt neat.txt thefourthfile.txt
$ ls
something
$ cp -r something newplace
$
$ rm something/awesome.txt
$ rmdir something
$ rm -rf newplace
$ ls
$
```

16.1.2 Windows

```
> cd temp
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            newplace
d----        12/22/2011   4:52 PM            something
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt
-a---        12/22/2011   4:49 PM          0 uncool.txt


> rm uncool.txt
> ls


    Directory: C:\Users\zed\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            newplace
d----        12/22/2011   4:52 PM            something
-a---        12/22/2011   4:49 PM          0 iamcool.txt
-a---        12/22/2011   4:49 PM          0 neat.txt
-a---        12/22/2011   4:49 PM          0 thefourthfile.txt


> rm iamcool.txt
> rm neat.txt
> rm thefourthfile.txt
> ls


    Directory: C:\Users\zed\temp
```

```


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        12/22/2011   4:52 PM            newplace
d----        12/22/2011   4:52 PM            something


> cp -r something newplace
> rm something/awesome.txt
> rmdir something
> rm -r newplace
> ls
>
```

16.2 You Learned This

Here we clean up the files from the last exercise. Remember when I had you try to rmdir on a directory with something in it? Well, that failed because you can't remove a directory with files in it. To do that you have to remove the file, or recursively delete all of its contents. That's what you did at the end of this.

17 Exiting Your Terminal (exit)

17.1 Do This

17.1.1 Linux/OSX

```
$ exit
```

17.1.2 Windows

```
> exit
```

17.2 You Learned This

Your final exercise is how to exit a Terminal. Again this is very easy, but I'm going to have you do more.

17.3 Do More

For your last set of exercises I'm going to have you use the help system to look up a set of commands you should research and learn how to use on your own.

17.3.1 Here's the list for Unix:

- xargs
- sudo
- chmod
- chown

17.3.2 For Windows look up these things:

- forfiles
- runas
- attrib
- icacls

18 Command Line Next Steps

You have completed the crash course. At this point you should be a barely capable shell user. There's a whole huge list of tricks and key sequences you don't know yet, and I'm going to give you a few final places to go research more.

18.1 Unix Bash References

The shell you've been using is called Bash. It's not the greatest shell but it's everywhere and has a lot of features so it's a good start. Here's a short list of links about Bash you should go read:

18.1.1 Bash Cheat Sheet

http://cli.learncodethehardway.org/bash_cheat_sheet.pdf
created by Raphael and CC licensed.

18.1.2 Reference Manual

http://www.gnu.org/software/bash/manual/bashref.html

18.2 PowerShell References

On Windows there's really only PowerShell. Here's a list of useful links for you related to PowerShell:

18.2.1 Owner's Manual

http://technet.microsoft.com/en-us/library/ee221100.aspx

18.2.2 Cheat Sheet

http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=7097

18.2.3 Master PowerShell

http://powershell.com/cs/blogs/ebook/default.aspx