

...KLTDSQNFDEYMKALGVFATRQVGLNLYLVSQEGGKV...

Protein Sequence

Computational methods



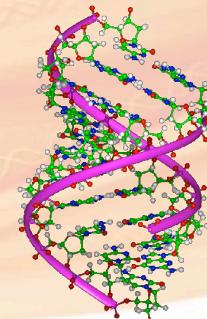
Protein Structure Model

Bioinformatics Computational Methods 1 - BIOL 6308



September 10th 2013

<http://155.33.203.128/cleslin/home/teaching6308F2013.php>



Last Time

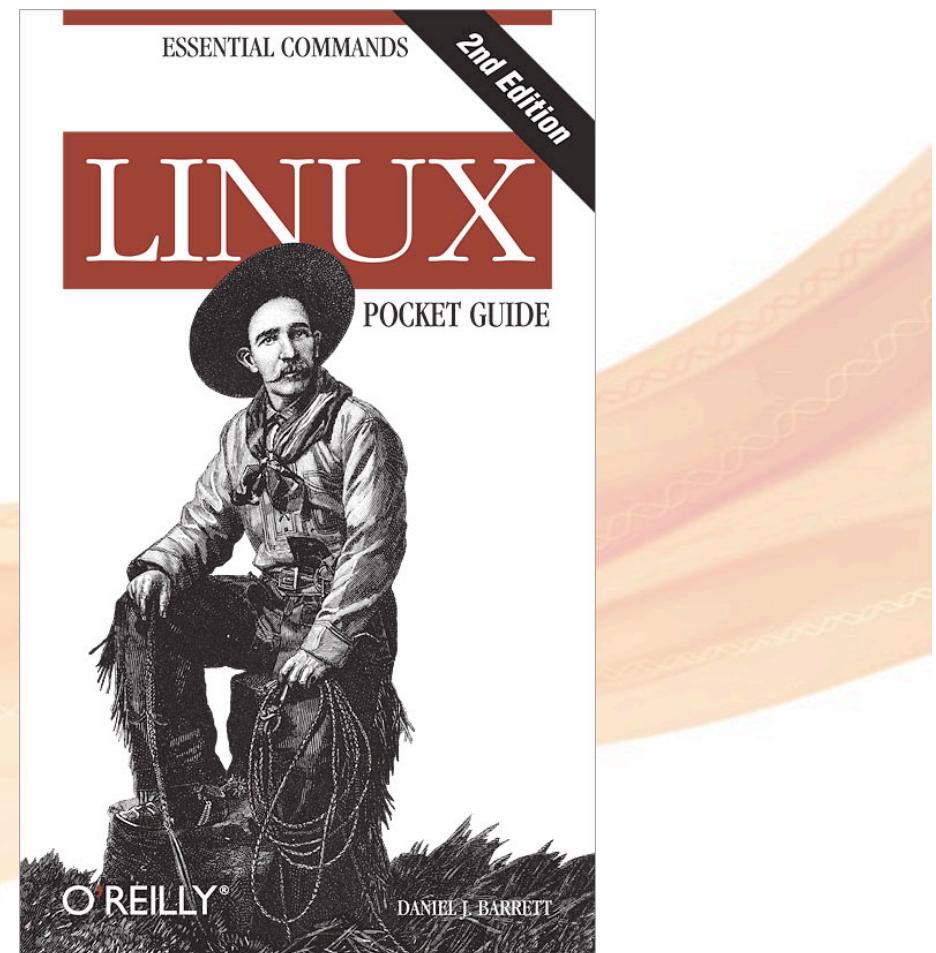
- Goals for Course
- Introduction Bioinformatics
- Units and Sources of Biological Data for bioinformatics
- Bioinformatics Methods and Algorithms
 - Types of the Data
 - Data storage, retrieval and visualization
 - Data mining
 - General Types of “Informatics Techniques in Bioinformatics”

How To Work With UNIX/Linux & Introduction of The Shell

Great Pocket Guide

If you use Linux in your day-to-day work, this popular pocket guide is the perfect on-the-job reference. The second edition has expanded from Fedora-only coverage to **distro-neutral**, with practical information on a wider range of commands requested by readers.

Linux Pocket Guide **provides an organized learning path for Linux use**, rather than Linux programming and system administration. You'll find options for the most **useful commands, grouped by functionality**. For novices who need to get up to speed on Linux use, and experienced users who want a concise and functional reference, this guide provides quick answers.



<http://shop.oreilly.com/product/0636920023029.do>

Definitions - Operating System (OS)

- A program or a software that governs the functioning of other programs
- Interface between User and the Hardware
- Allocates resources for tasks
- Allocates tasks to programs
- Manages space and time
- Controls the devices

Definitions - Types of OS

- Tasks
 - Uni-tasking OS
 - Multi-tasking OS
- Users
 - Single User OS
 - Multi User OS
- Processing
 - Uni-processing OS
 - Multi-processing OS
- Time-sharing OS

Definitions - Kernel

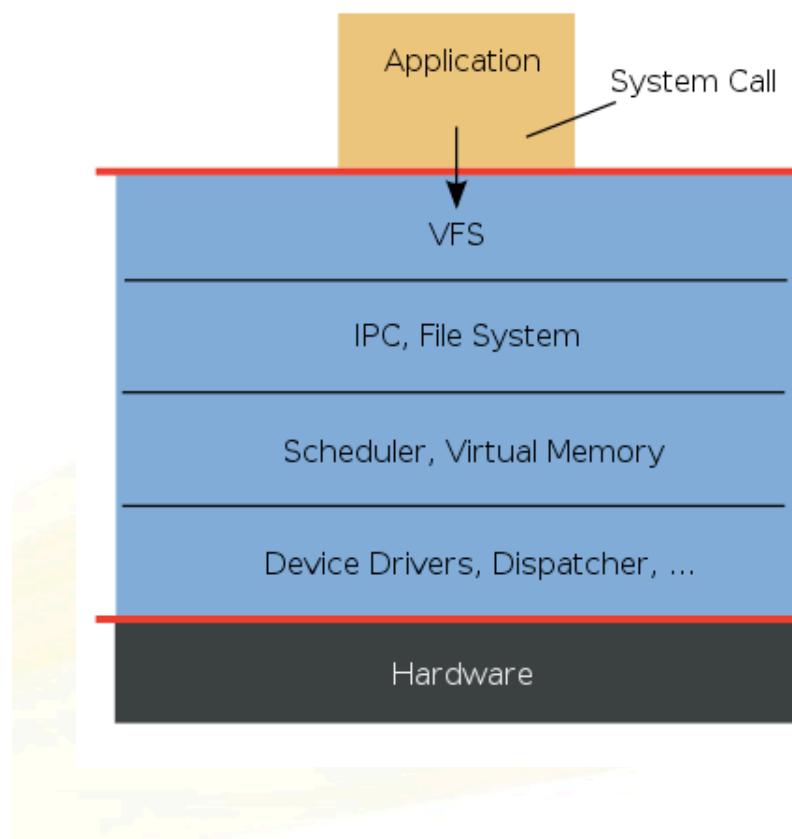
- Core or nucleus of an OS
- OS consists two parts:
 - The Kernel space (privileged mode) and the user space (mode) (unprivileged mode)
- Interacts with hardware
 - Bridge between **applications** and the actual data processing done at the **hardware level**
- First program to get loaded when the system starts and runs till the session gets terminated
- Different from BIOS which is hardware dependent
- Kernel is software dependent

Definitions - Kernel Types

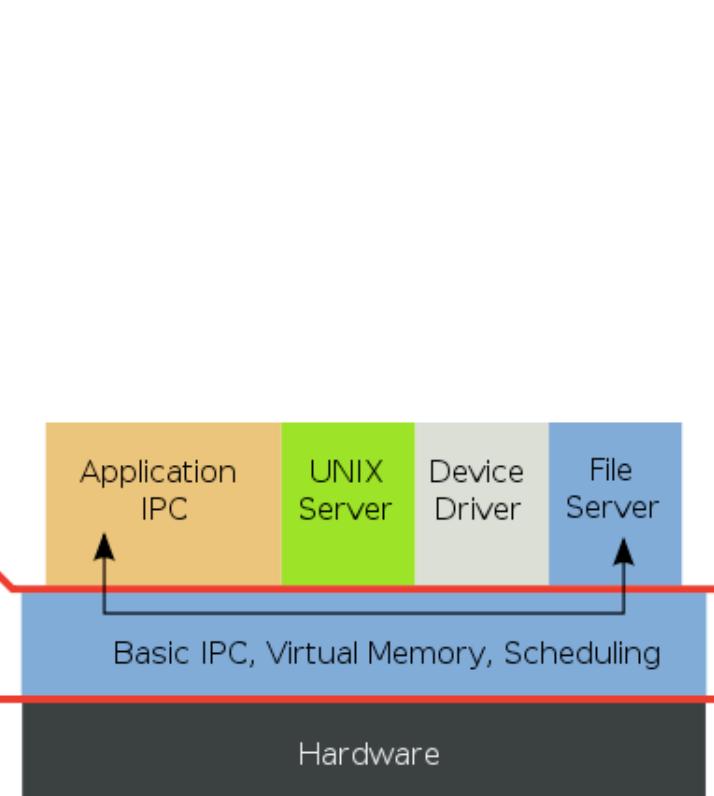
- Monolithic
 - A single address space is used to execute OS code
 - All OS related code are stuffed in a single module
 - Available as a single file
 - Advantage : Faster functioning
- Microkernel
 - Kernel itself has minimal OS-to-hardware demands
 - OS components are isolated and run in their own address space
 - Device drivers, programs and system services run outside kernel memory space
 - Supports modularity
 - Small in size

FYI - Mono-Micro Kernels

**Monolithic Kernel
based Operating System**



**Microkernel
based Operating System**



History of Linux

- Multics – 1964
- Unics – 1969
- Minix – 1990
- Linux – 1991

FYI - Multics

- Multiplexed Information and Computing Service
- Written in 1964
- Timesharing OS
- Last version was shut down on October 30, 2008
- Monolithic kernel

FYI - Unics

- Uniplexed Information and Computing System
- Later renamed as UNIX
- Written in 1969
- Ken Thompson and Dennis Ritchie were among the developers
- Multi user, Multi tasking and timesharing
- Monolithic kernel

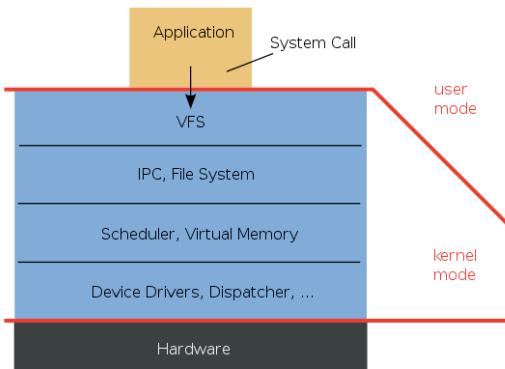
FYI - Minix

- Minimal UNIX
- Tanenbaum developed this OS
- Mainly for educational purpose
- UNIX like OS, implemented with Micro kernel. So the name Minix

Linux

- Developed in 1991 by Linus Torvalds
- Multi tasking
- Time sharing
- Monolithic kernel
- Latest stable version of linux kernel – **3.10.11**, released 2013-09-08

Monolithic Kernel
based Operating System



Linux OS

Main components of Linux operating system

GUI:

Gnome KDE
 X.org

gcc

GNU coreutils

LAMP:
Apache
MySQL
Perl/PHP/Python

Net:
sshd
inetd

bash

GNU C Library

other libraries

SCI device files

Linux kernel

sockets

processes

file systems

protocols

memory management

drivers and modules

computer hardware

Linux Distributions

- Redhat
- **Fedora**
- Debian
- Novell's SUSE Linux
- Ubuntu
- Mandrake
- Live CDs

GNU/Linux

- Only the Kernel is called by the name Linux
- The rest are the tools developed under GNU Project
- Hence the name GNU/Linux



But everyone calls the OS Linux, including myself!

UNIX vs. Linux

- UNIX was the predecessor of Linux
- Linux is a variant of UNIX
 - So is Mac OS X, so much of this tutorial applies to Macs as well!
- Linux is open source (free)
- Most of the machines doing Bioinformatics in industry are running the Linux OS
- It's as stable as UNIX (almost never crashes)
- These characteristics make it an ideal OS for programmers and bioinformaticians

Linux Overview

- Why UNIX/Linux?
 - Control
 - Commands often provide complete access to the system and its devices
 - Ability to automate bioinformatics tasks
 - Remote access
 - Flexibility
 - Commands are just programs
 - Commands have common interface to allow interoperation with other commands
 - The Linux shells provide the “glue” for this
 - Don’t have to learn "where everything is"
 - Reliability
 - Commands are typically lightweight since they typically do little more than invoke operating system calls
 - Doesn't waste computer resources on graphics
- POWER

What Do I Mean By Power!

- Large memory servers (LMS)
 - Like the one we have!
- Parallel Computing Architectures
- Collection of processing elements that can communicate and cooperate to solve large problems fast
- Must learn Linux



General Tips

- UNIX/Linux/Potentially OSX is case sensitive!!
 - myfile.txt and MyFile.txt do not mean the same thing
 - I like to use capital letters for directory names - it puts them at the top of an alphabetical listing
- **Every program is independent**
 - Core operating system (**kernel**) manages each program
 - As a distinct process
 - With its own little chunk of dedicated memory
 - If one program runs into trouble, it dies, but does not affect the kernel or the other programs running on the computer
 - That's nice!

Our Hardware: Fisher!!

- **Server specs:** Dell PowerEdge R820 HPC server
- Four Intel® Xeon® E5-4650 2.70 GHz, 20M Cache, 8.0GT/s QPI, Turbo, 8C, 130W
 - 32 cores, 62 threads!
- 64GB total - 4 x 16GB RDIMM, 1600MT/s, Low Volt, Dual Rank, x4 Data Width
- Hard Drives: (RAID 1 = 2 x 146GB 15K RPM SAS 6Gbps 2.5in) + (RAID 5 = 4 x 900GB 10K RPM SAS 6Gbps 2.5in)
- **Server URL:** fisher.neu.edu



Speed!

- Simple benchmark with Perl

Benchmark: timing 100000 iterations of dewey, huey, louie...

chopin (old server)

dewey: 205 wallclock secs (204.37 usr + 0.02 sys = 204.39 CPU)

@ **489.26/s** (n=100000)

huey: 211 wallclock secs (210.59 usr + 0.00 sys = 210.59 CPU)

@ **474.86/s** (n=100000)

louie: 165 wallclock secs (165.74 usr + 0.00 sys = 165.74 CPU)

@ **603.35/s** (n=100000)

fisher

dewey: 87 wallclock secs (87.05 usr + 0.00 sys = 87.05 CPU)

@ **1148.77/s** (n=100000)

huey: 100 wallclock secs (100.24 usr + 0.00 sys = 100.24 CPU)

@ **997.61/s** (n=100000)

louie: 77 wallclock secs (77.02 usr + 0.00 sys = 77.02 CPU)

@ **1298.36/s** (n=100000)

Details...

- **Student Username:** First letter of first name + last name
 - Ex: John Smith = jsmith (no spaces or underscores)
- **Student Password:** Emailed to you by Stefan
 - Must change within 1 week otherwise password expires and account locks
 - \$ passwd (to change password)
 - All new passwords **MUST** contain a capital letter, at least one number, and **MUST** be at least eight characters long
- **Student '/home/' limit:** 25GB (home folders are backed up)
- **'/scratch'** - working directory
 - If you use this, please make your own subdirectory and keep all your files within there
 - 'chmod 700' your subdirectory so that other people cannot see/alter your files
 - **'/scratch' is not backed up!!!** Anything over 30 days old may be deleted without notice
- **Available space on '/scratch'** = 1.2TB (expanding to 3TB soon!!)
 - No space limitation on '/scratch'
 - Please be mindful that this is a shared computing resource and use only what you need
 - Delete files once you are done!!!

Firewall

- With very few exceptions, most connection are not allowed
- Can make downloading to the server a pain, but IT leaves us with no choice
- Easiest to download files locally and then 'scp' them to 'fisher'
- If you fail a log in >6 times, your IP will be blocked from accessing the server for 3 days. (email your username and IP to s.kaluziak@neu.edu to remove from blacklist)
- BioPerl access to outside database works

Restrictions

- **Restrictions:** Everyone using the server MUST adhere to Northeastern's AUP policy
 - http://www.northeastern.edu/infoservices/?page_id=97
 - Fisher is for research/educational purposes only
- Play nice!

List of Installed Programs

- **All installed programs are in PATH Environment Variable**

- PATH=\$PATH:/usr/local/trinityrnaseq_r2012-10-05
- PATH=\$PATH:/usr/local/velvet_1.2.08
- PATH=\$PATH:/usr/local/MUMmer3.23
- PATH=\$PATH:/usr/local/fasta-36.3.5e/bin
- PATH=\$PATH:/usr/local/clustalw-2.1
- PATH=\$PATH:/usr/local/bowtie2-2.0.5
- PATH=\$PATH:/usr/lib64/openmpi/bin
- PATH=\$PATH:/usr/local/lib
- PATH=\$PATH:/usr/local/FastX_ToolKit_0.0.13/
- PATH=\$PATH:/usr/local/ncbi-blast-2.2.27+/bin/
 - (Installed the NR, SwissProt, TrEMBLE, TAXID)
- PATH=\$PATH:/usr/local/stacks-0.9999/scripts

What is PATH? We'll learn about it soon, but here is a start:
http://www.cs.purdue.edu/homes/cs348/unix_path.html

The Following are Installed:

- In '/usr/local/bin':
 - mrbayes_3.2.1
 - muscle
 - muscle3.8.31
- Installed with yum
 - samtools
 - bwa
 - bowtie

Perl and Python

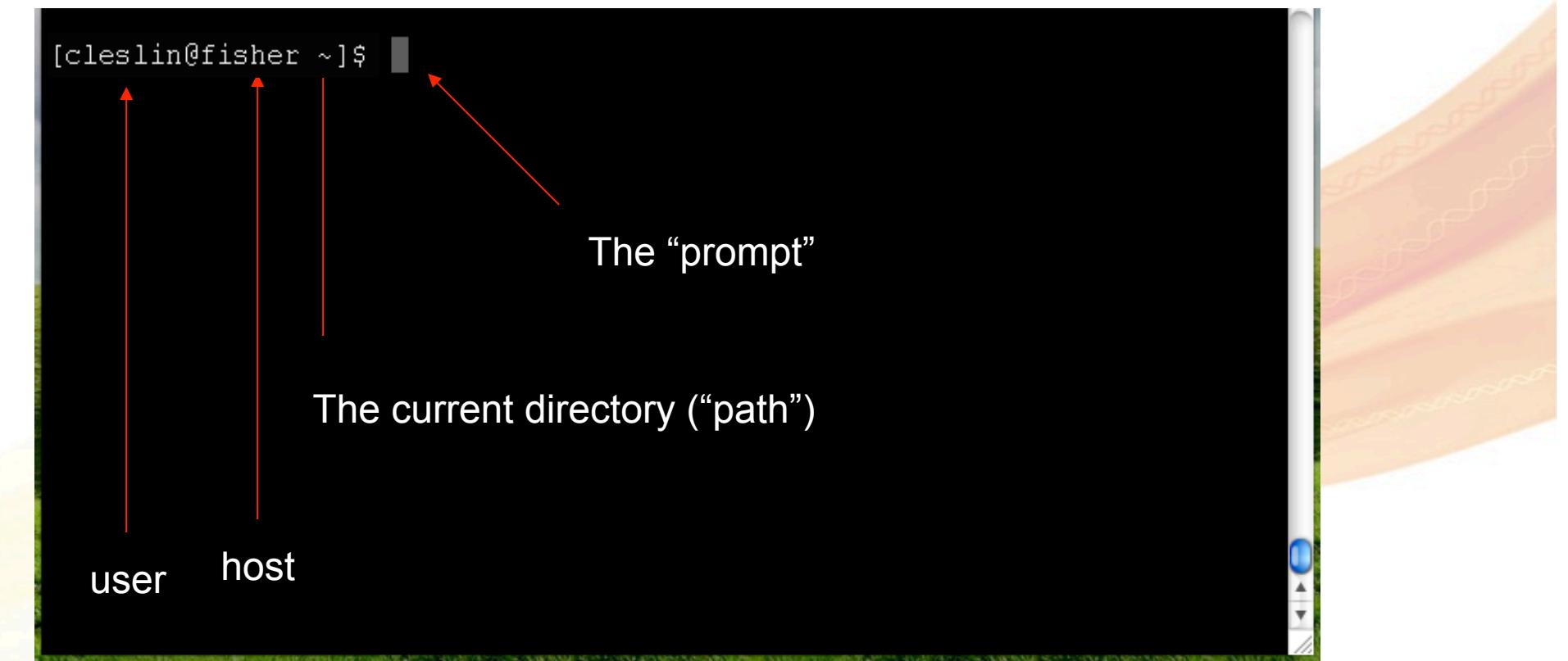
- Perl
 - BioPerl
 - BioPerl-run
- Python
 - Python-devel
 - BioPython
 - NumPy
 - SciPy

Accessing fisher Server

- ssh
 - Secure Shell
 - Protocol (in some cases, a program)
 - Default port number for ssh is 22
 - Opens terminal

Connecting to Our Linux system

- Open up a terminal:



chopin was our old server (you might come across it in lectures/labs
! fisher is our new server

UNIX: The Command Line

- Accessing UNIX through a terminal
 - ssh <user>@fisher.neu.edu
 - Secure
 - Data is encrypted over “the wire”
 - Its what we use
 - Need something on Windows
 - » Suggest: Putty or OpenSSH
 - Terminal provided on MAC
- Show who else is logged in
 - w or who
 - finger <user>
- Logout!
 - exit

Change Your Password

- After your original login, change your password!
- This is very, very, very important!
- passwd (change your user password often)
 - Use complex non-dictionary words
 - Use at least one 0-9 and include one capital letter

Linux: Accessing Documentation

- Commands **generally documented** using the command `man`
 - man pages are subdivided into various *sections*
 - Example: Documentation of the `man` command
`man man`
 - Example: Documentation of the `time` command
`man time`
- `man` will present the manual page of the specified entry using `more` or `less`
 - default is `less`, but can be changed
 - `less` presents a screen-full at a time. 'spacebar' moves forward, 'b' moves backward, 'q' quits, '?' helps
- Most commands have HTML pages on the WWW
- Also use `whatis`

```
$whatis cp
```

```
cp []          (1) - copy files and directories
```

Filename Extensions

- Most UNIX filenames start with a lower case letter and end with a dot followed by one, two, or three letters: myfile.txt
 - Just a common convention and is not required
 - Possible to have additional dots in the filename
- The part of the name following the dot is called the “extension”
- Extension is often used to designate the type of file
 - .sh
 - .pl
 - .java
 - .py
 - .fasta

Some Common Extensions

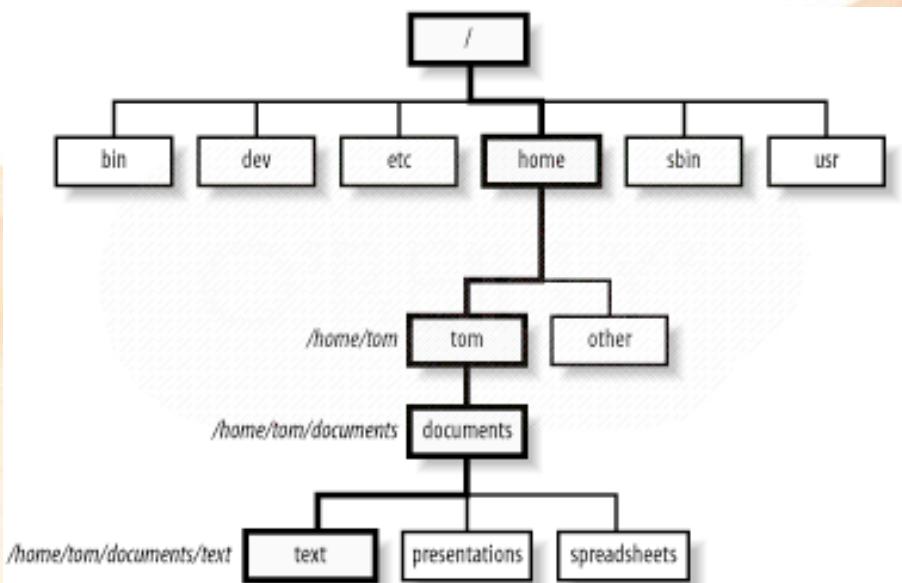
- By convention:
 - Files that end in:
 - .txt are text files
 - .c are source code in the "C" language
 - .html are HTML files for the Web
 - Compressed files have the .zip or .gz extension
- **UNIX does not require these extensions (unlike Windows), but it is a sensible idea and one that you should follow**

Linux: Getting Around the Filesystems

- Linux files are organized just like they are with PC's and MAC's
- Files are contained in collections of Directories
- Directories may contain other Directories
- There are no drive letters!!
- The “top level” directory is called the “root” directory and is referred to by “/”
 - You have write access to /home/your_user_name
 - Have been given 25 GB of space to use
- Current directory is referred to by “.”
- Directory one level up is referred to by “..”
- More dots don't get you more levels up

Linux Directory Structure

- The format of the Linux Filesystem may be strange at first
- Most time work in
 - `/home/username`
- Most of the applications you use will be in
 - `/usr/bin`
 - `/usr/local/bin`



Don't worry, you don't have to learn all this to use Linux directories

Working Directory

- The **current directory** in which you are working
- `pwd` command: outputs the absolute path (more on this later) of your working directory
- Unless you specify another directory, commands will assume you want to operate on the working directory

Home Directory

- A special place for each user to store personal files
- When you log in, your working directory will be set to your home directory
- Your home directory is represented by the symbol ~ (tilde)
- The home directory of “user1” is represented by ~user1

Linux Directory Structure (cont.)

- Important Directories
- **/bin** This contains files that are essential for correct operation of the system. These are available for use by all users
- **/home** This is where user home directories are stored
- **/var** This directory is used to store files which change frequently, and must be available to be written to
- **/etc** Various system configuration files are stored here
- **/dev** This contains various devices as files, e.g. hard disk, CD-ROM drive, etc.
- **/sbin** Binaries which are statically linked
- **/tmp** Temporary files

File System Commands

- **pwd** - report your current directory
- **cd <to where>** - change your current directory
- **ls <directory>** -list contents of directory
- **cp <old file> <new file>** - copy
- **mv <old file> <new file>** - move (or rename)
- **rm <file>** -delete a file
- **mkdir <new directory name>** -make a directory
- **rmdir <directory>** -remove an empty directory
- **df** - see how much space is on the filesystem
- **du -a** -see how much space you are taking
- **scp** - copy a file over a secure connection to a server

Linux: Getting Around

- Commands to navigate the directories:

- change directories

```
cd /data/METHODS/Fall/
```

```
cd ~your_user_name
```

```
cd /data/METHODS/Fall/LAB1
```

- **cd** # go back home

- **mkdir test**

- **rmdir test** # if directory is empty you can get rid of it like this

- copy files and directories

```
cp /data/METHODS/Fall/test/junk1.txt . ## copy into your current directory
```

```
cp /data/METHODS/Fall/test/JUNK . #try and copy this directory to your home directory (wont work!!)
```

```
cp -r /data/METHODS/Fall/test/JUNK . #if the directory contains things you can't copy w/o -r
```

Linux: More on Getting Around

- More Commands to navigate the directories:

- **mv JUNK/junk2.txt .** #move the document into your working directory
- **cp junk2.txt junk3.txt** #create a copy
- remove files and directoreis
rm junk3.txt
rm -r JUNK
- find
cd /tmp
find . -ls

Getting Recursive

- Recursive means everything in a directory
 - copy a directory and its contents:
 - `cp -r <directory>`
 - `cd`
 - `cp -r /data/METHODS/Fall/test/JUNK`
 - remove a directory and its contents:
 - `rm -r <directory>`
 - `rm -r /home/user_name/JUNK`
 - Listing the contents of a directory and all subdirectories
 - `ls -R <directory> or ls -lR <directory>`

The User

- Normal user and super user (su)
- In Linux and OSX systems, there is one special user for the administrator
- Can do anything (su)
- **su** called **root** or **superuser**
- **root** conventional name of the user who has all rights or permissions (to all files and programs) in all modes (single- or multi-user)
- Under Linux and OSX you use command called:
 - su
- **Do not do work in this mode**
- **Do not do work in this mode**
- **Do not do work in this mode**
 - I've warned you!!!!

With great power comes great responsibility!!!!

File Names

- Up to 256 characters in length
- Can start with or be all numbers
- Can use period(.) underscore(_), dash(-) or commas(,)
- Case sensitive (PC usually not) (Mac has to be setup to be)
- As a general rule: Do not include spaces in file names

File Permissions

- Linux is a proper multi-user environment
- Security of user and system data is very important
- There are 3 kinds of people in the world: you (user), your friends (group) and everyone else (others)
- Each sort of person may or may not be able to read, write, or execute a file
- “r” means “read only” permission
- “w” means “write” permission
- “x” means “execute” permission
 - In case of directory, “x” grants permission to list directory contents

```
>ls -l junk1.txt
-rw-r--r-- 1 cleslin cleslin 5 Sep 23 18:02 junk1.txt
```

File Permissions

```
-rw-rw-r--  1 user1 user1   169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--  1 user1 user1    92 Aug 30 11:54 ACTG.pl
-rw-rw-r--  1 user1 user1    21 Aug 30 12:23 data.dat
-rw-rw-r--  1 user1 user1    42 Aug 30 12:22 hello_world.pl
-rw-rw-r--  1 user1 user1    24 Aug 30 12:23 input.txt
-rw-rw-r--  1 user1 user1    50 Aug 30 13:13 lines.txt
drwxrwxr-x  2 user1 user1  4096 Aug 30 13:19 new_directory
home:~/linux_tutorials$ █
```

User (you)

File Permissions

```
-rw-rw-r--  1 user1 user1   169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--  1 user1 user1    92 Aug 30 11:54 ACTG.pl
-rw-rw-r--  1 user1 user1    21 Aug 30 12:23 data.dat
-rw-rw-r--  1 user1 user1    42 Aug 30 12:22 hello_world.pl
-rw-rw-r--  1 user1 user1    24 Aug 30 12:23 input.txt
-rw-rw-r--  1 user1 user1    50 Aug 30 13:13 lines.txt
drwxrwxr-x  2 user1 user1  4096 Aug 30 13:19 new_directory
home:~/linux_tutorials$
```

Group

File Permissions

```
-rw-rw-r-- 1 user1 user1 169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r-- 1 user1 user1 92 Aug 30 11:54 ACTG.pl
-rw-rw-r-- 1 user1 user1 21 Aug 30 12:23 data.dat
-rw-rw-r-- 1 user1 user1 42 Aug 30 12:22 hello_world.pl
-rw-rw-r-- 1 user1 user1 24 Aug 30 12:23 input.txt
-rw-rw-r-- 1 user1 user1 50 Aug 30 13:13 lines.txt
drwxrwxr-x 2 user1 user1 4096 Aug 30 13:19 new_directory
home:~/linux_tutorials$
```

other = “The World”

For more info: <http://catcode.com/teachmod/>

Changing File Permissions

- make a file readable to your friends:
\$ chmod g+r <filename>
- make a program executable for (user) you
\$ chmod u+x test.pl
- Make a file writable to your group
- **\$ chmod g+w <filename>**
- Take permissions away, just the opposite
\$ chmod g-r <filename>
- change who owns a file:
`chown <user> <filename>`
\$ chown cleslin test.pl
\$ chown -R cleslin <directory>
- change to which group the file belongs:
\$ chgrp <group> <filename>

Types of Files

- Plain (- in the first bit)
 - Most files
 - Includes binary and text files
- Directory (d)
 - A directory is actually a file
 - Points to another set of files
- Link (l): A pointer to another file or directory, check out the next slides!

```
-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$ █
```

Linking Files

- ln – Creating links between files
 - ln file1 file2
 - Difference between copying files and linking files
 - cp src dst
 - Contents of src will be present in dst
 - Changing content in src will not affect contents of dst and vice versa
 - ln src dst
 - Contents of src will be present in dst
 - Changing content in src or dst will get reflected in the other file
 - **Most likely want to create a soft link, so use -s**

<http://www.cyberciti.biz/tips/understanding-unixlinux-symbolic-soft-and-hard-links.html>

Symbolic Links

- use `ln -s <old file> <second name>` to create a symbolic link to a file/directory/program

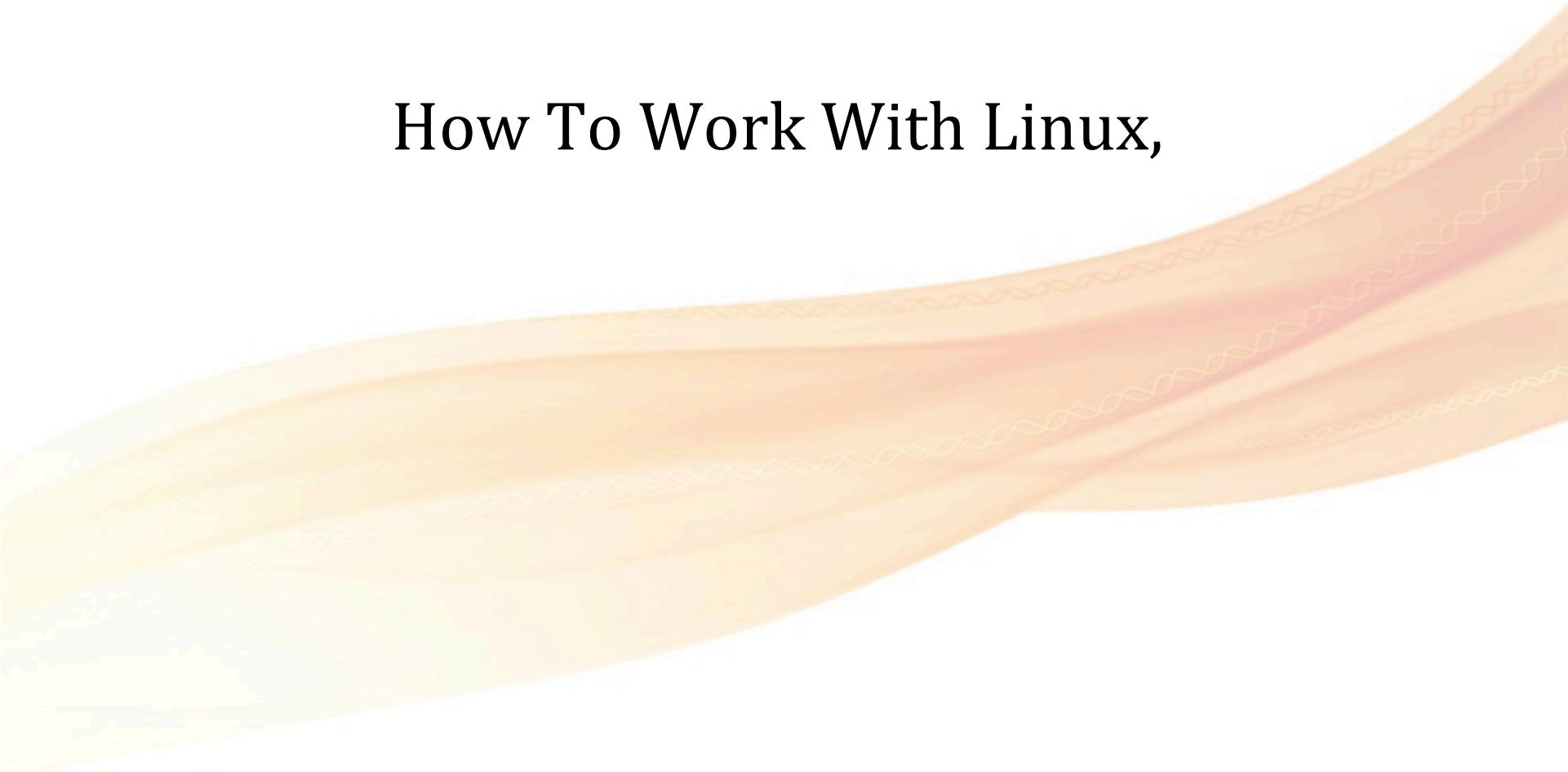
```
$ cd JUNK
$ ln -s /home/your_user_name/ test
$ ls -l

rw-r--r-- 1 cleslin staff 8 Sep 23 18:04 junk1.txt
-rw-r--r-- 1 cleslin staff 8 Sep 23 18:04 junk2.txt
-rw-r--r-- 1 cleslin staff 8 Sep 23 18:04 junk3.txt
lrwxr-xr-x 1 cleslin staff 15 Sep 23 18:28 test -> /home/your_user_name/
```

- Like creating a shortcut
- Symbolic links can be used as if it were its target

Archives / Compressing

- tar is just as it sounds, it “tars” files together
 - Usually used in conjunction with compressors such as gzip (file.tar.gz) or bzip2 (file.tar.bz2)
- Creating an archive
 - `tar -czf myarchive.tar.gz file1 file2 directory1`
- List files in an archive
 - `tar -tzf myarchive.tar.gz`
- Extract an archive
 - `tar -xzf myarchive.tar.gz`



How To Work With Linux,

Relative & Absolute Path

- Path means a position in the directory tree
- To express a path, you can use relative path or absolute path
- In relative path expression, the path is not defined uniquely, depends on your current path
 - A **relative** path is a path relative to the working directory of the user or application
- In absolute path expression, the path is defined uniquely, does not depend on your current path
 - A **absolute** path is a path that points to the same location on one file system regardless of the working directory or combined paths

Linux absolute path

/home/cleslin/public_html/cgi-bin

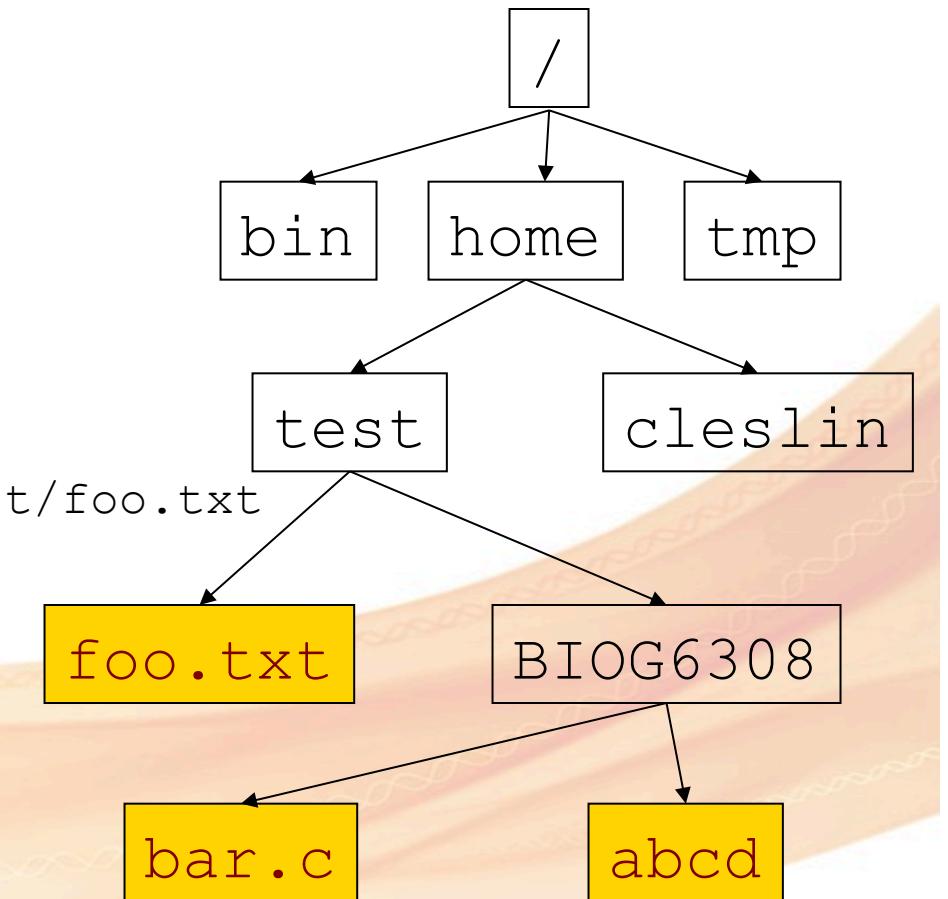
Linux non absolute path (relative path)

../public_html/cgi-bin

<http://linux.about.com/cs/linux101/g/relativevsdtabs.htm>

Path Names

- Separate directories by /
- Absolute path
 - start at root and follow the tree
 - e.g. /data/METHODS/Fall/test/foo.txt
- Relative path
 - start at working directory
 - .. refers to level above
 - . refers to working dir
 - If /home/test/BIOG6308 is working dir, all these refer to the same file
 - ../foo.txt ~/foo.txt ~test/foo.txt



touch

- touch – easiest way to create new, empty files
- Also used to change the timestamps (i.e., dates and times of the most recent access and modification) on existing files and directories

```
$ ls -l junk1.txt
```

```
-rw-r--r-- 1 cleslin staff 5 Sep 23 18:02 junk1.txt
```

- Each file has a date stamp of when it was modified
- Use touch to set the timestamp to the current clock

```
$ touch junk1.txt
```

```
-rw-r--r-- 1 cleslin staff 5 Sep 23 18:25 junk1.txt
```

- touch create the file if it didn't exist
 - Useful for a program to create something showing it has finished
 - You can only touch a file to which you can write
- In contrast cp and mv, it does not automatically overwrite
- Merely changes last access times for such files to the current time

Wild Characters

- Characters that can be used to match a number of different character combinations
- (?) - represents 1 character
 - ex. Lab?.pl could match:
 - Lab3.pl
 - Lab8.pl
 - LabN.pl
- (*) - represents more than 1 character
 - ex. Lab*.pl could match:
 - Lab3.pl
 - Lab300000.pl
 - LabA354.pl

Linux: More Useful Commands

echo

```
$ echo "$HOME is where I want to be"
```

locate

find where a file/directory is

printenv

Display all environment variables

head

```
$ head -n 5 file
```

first 5 lines of output

tail

last 8 lines of output

```
$ tail -n 8 file
```

find

find all empty files current

directory down

```
$ find . -size 0 -print
```

More on find

```
$ find Looks for a file named "game" starting at the root directory
```

- name makes it case sensitive
- use name find something regardless of case

```
$ find / -name game
```

- Find every file under the directory /home/cleslin ending in "stat"

```
$ find /home/cleslin -name *stat
```

- Find every file under the directory /home/cleslin that was modified more than 60 days ago. ad

```
$ find /home/cleslin/ -mtime +60
```

- Find every file under the directory /home/cleslin that was modified last 24 hours

```
$ find /home/cleslin -mtime 0
```

The cat Command

- The arguments to cat are concatenated together and displayed on stdout. To view a file:

```
$ cat /data/METHODS/Fall/test/test.fasta
```

- if no arguments, cat put on stdout whatever you type on stdin, so this does the same thing:

```
$ cat < /data/METHODS/Fall/test/test.fasta
```

- if you want to concatenate two files, then you need to redirect the output

```
$ cat /data/METHODS/Fall/test/test.fasta /data/METHODS/Fall/test/test2.fasta >test3.fasta
```

- Not limited to one file

```
$ cat junk1.txt junk2.txt junk3.txt >BigJunk123.txt
```

Sorting

\$ sort Sorts using the first field of each line.

-n Sorts considering the numeric value of the strings

-k3 Sorts using the third field of each line

-rnk3 Sorts in reverse order, using the numeric
value of the third field

-u Suppress all but one line in each set of lines with
equal sort fields

sort -nk2 /data/METHODS/Fall/test/sort.txt

<http://lowfatlinux.com/linux-sort.html>

Like ssh, There is ...scp

- ssh is used to securely log onto remote servers

```
$ ssh [username]@[hostname]
```

- scp is used to copy files to/from remote systems, syntax is similar to cp:

```
$ scp [local path] [username]@[hostname]:[remote file path]
```

- So if you have another Linux box or a Mac, Try it

- Copy to: ON my Mac

```
$ scp hello.txt username@fisher.neu.edu:/home/username
```

hello.txt should be in your working directory

- Copy from: ON my Mac

```
$ scp username@fisher.neu.edu:/home/username/foo.txt .
```

foo.txt should be in your home directory

Don't forget the trailing " . "

Linux Pipelines

- Pipelines are created | joining the output of | one program | with the next
 - Pipes take the output of the first program
 - Feed that output into the input of the next program
- Also sometimes known as “filters”
- Examples:

```
$ last | less
```

```
$ last | grep '^cleslin' | less
```

```
$ grep 'error' /data/METHODS/Fall/test/something.out | tail -n 1
```

Linux Redirection: Lesser and Greater

- Two types of output
 - Standard output (when program produces normal output)
 - Standard error (when something breaks in the program)
- `>filename` redirects just standard output

```
$ abcd >junk  
-bash: abcd: command not found  
$ abcd >junk 2>stderr
```

- `>&filename` redirects the standard output and error to the file called *filename*:

```
$ \time ls >junk  
0.00 real          0.00 user          0.00 sys  
  
$ \time ls >&junk
```

- By default, `>` will overwrite existing files
 - Can append to a file with `>>`
- Use `<` to redirect a file to a command's *standard input*

```
$ cat /data/METHODS/Fall/test/calculation.txt  
(3+2) *8  
$ bc < /data/METHODS/Fall/test/calculation.txt
```

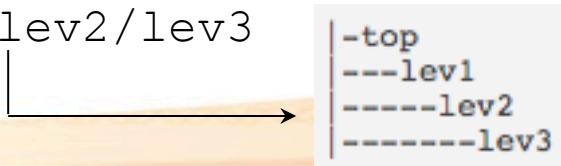
Additional Linux commands

- date print out current date/time
- diff Report differences between two files
- sdiff Report differences side-by-side
- wc Show number of lines, words in a file
- gzip Compress a file
- gunzip Uncompress it
- dos2unix Convert windows based files to unix

Make Parent Directories As Needed

- Familiar with the mkdir command to create a single directory
 - adding -p option:
 - create multiple directories with a single command
 - Most basic form, it can be used to create a single directory, with a single child, grand child and so on

```
$ mkdir -p top/lev1/lev2/lev3
```



FYI - Make Complete Directory Structures

- Or you can make an entire Directory Structures

```
$ mkdir -p coolapp/{docs/{en_uk/{html,text, pdf}, en_us/
  {html, text, pdf}}, includes/{functions,language/
  {en_uk, en_us}, classes}, templates, tmp}
```



```
-coolapp
---docs
    ---en_uk
        -----html
        -----pdf
        -----text
    ---en_us
        -----html
        -----pdf
        -----text
---includes
    ---classes
    ---functions
    ---language
        -----en_uk
        -----en_us
---templates
---tmp
```

Linux Command Line Tricks

- Filename Completion (tcsh, bash)
 - \$ ls /tmp/o<TAB>
- Command line history
 - \$ history | less
 - spacebar - go forward
 - b - go back
 - shift-f go to the end, cntrl-c to get out
 - q - quit
- \$ grep 'what you're looking for' filename | less
 - \$ grep '^>' test.fasta | less #look at the header lines
 - \$ grep -c '^>' test.fasta ## how many times '>' in the file

History

```
$ history - output your command line history in two columns
```

- **First** being a number
- **Second** the command itself:

```
$ history | less
```

- The number allows for quick execution of past commands, when number is preceded with an ! exclamation mark
 - Say, execute fifth command in your history - simply run:

```
$ !5
```

- Double !! exclamation marks will simple run the previous command:

```
$ !!
```

Advanced History

- What if you can't remember a particular command, or only partially remember a long command?
- Manually looking through hundreds of lines can be a strain
- `grep` used to search for a term, before being piped through to `less`:
- Use the format:

```
$ history | grep '<searchterm>' | less
```
- Example:

```
$ history | grep 'ls' | less
```
- Return list of commands found in history containing “ls”
- The command number is also returned - easily be executed
 - How?
 - Preceding the number with an ! exclamation mark

Source Code

- In bioinformatics, most software is **free**, and **open source**
- You can download the actual instructions the programmer wrote
- This is great, because it means you can install these programs on almost any machine

The GNU install pragma

- GNU source code can be complicated to compile, so it comes with programs to help you
- There is a standard way to build and install GNU software

```
[user@server ~]$ tar -zxvf program.tar.gz
```

```
[user@server ~]$ cd program
```

```
[user@server ~]$ ./configure
```

```
[user@server ~]$ make
```

```
[user@server ~]$ make install
```

The shell

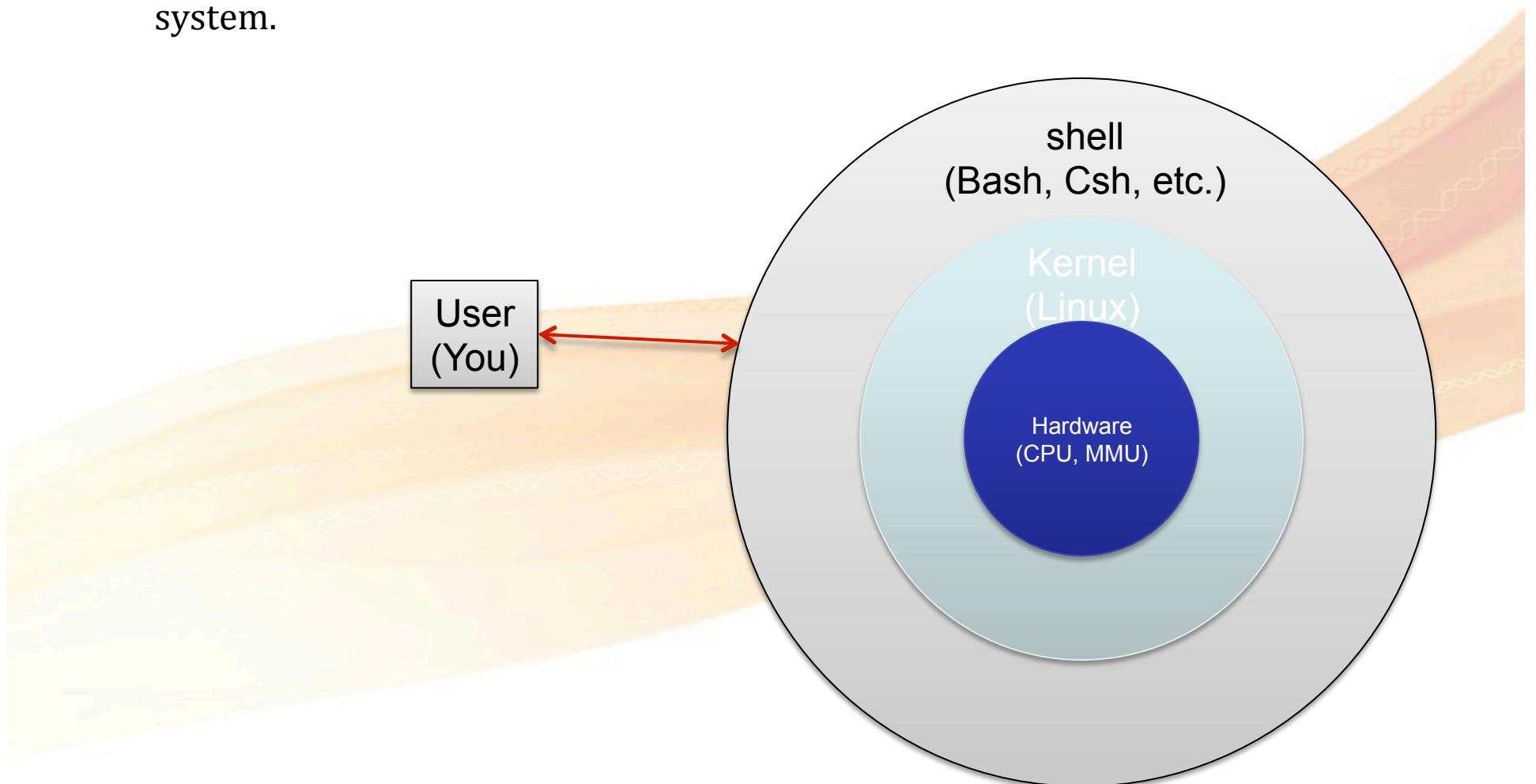


The shell

- You communicate with a UNIX computer through a command program known as a shell
 - Program that interacts with kernel
 - Bridge between kernel and the user
 - Command interpreter
- Interprets the commands that you type on the keyboard
- Many different shells available for UNIX computers
- Write simple programs (scripts) to automate many tasks using shell
- Say what?

What is a shell?

- A shell is a program that serves as the outermost interface to the operating system.



The shell

- Simply another program which provides a **basic human-OS interface**
 - **Command-Line Interface (CLI)**, with the full power of Linux multitasking available
 - Most common: [bash](#) (Bourne-again shell), where bash is a UNIX shell written for the GNU Project
- Bash default shell on most Linux systems
 - As well as on Mac OS X
 - Can be run on most UNIX-like operating systems
- A Linux shell is used when interfacing with the lower Linux boot levels (such as run levels 1 to 3) (<http://en.wikipedia.org/wiki/Runlevel>)
- Applications can be started from a shell, which can also be very useful a number of reasons

Linux: The shells

- Shells can run interactively or as a shell script
- Two main ‘flavors’ of Shells:
 - Bourne created what is now known as the standard shell: “sh”, or “bourne shell”
 - Syntax roughly resembles Pascal
 - Derivatives include “ksh” (“korn shell”) and now
 - The most widely used, “bash” (“bourne shell”).
 - One of the creators of the C language implemented the shell to have a “C-programming” like syntax
 - Called “csh” or “C-shell”
 - Another very popular shell is “tcsh”
 - To view the current shell that is being used, type:

```
$ echo $SHELL
```

Bash – Bourne Again Shell

- Supports Basic Regular Expressions
 - E.x. ls myfile.* or ls data4?
 - * matches 0+ of any character (wildcard)
 - ? matches 1 of any character
 - [abc] will match a, b, or c.
- Has a powerful scripting language
 - Bash scripting
 - You'll use this in Operating Systems

```
[cleslin@fisher letters]$ ls aa[ac]
aaa  aac
[cleslin@fisher letters]$ ls aa[a-c]
aaa aab aac
```

Linux: shell Path Basics

- Modifying *environment variables*

sh: PAGER=/usr/bin/less; export PAGER

bash: export PAGER=/usr/bin/less

- Execute an external command (sh)

```
$ somecommand
```

```
somecommand: command not found
```

```
$ echo $PATH
```

```
/usr/lib64/qt-3.3/bin:/usr/kerberos/bin:/usr/lib64/  
ccache:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/  
usr/sbin:/sbin:/usr/local/blast/blast-2.2.18/bin:/  
usr/local/fasta-35.4.7/bin:/home/cleslin/bin
```

```
$ pwd
```

```
/home/user_name
```

```
$ /data/METHODS/Fall/test/somecommand
```

```
Hello world!
```

```
$ cd /home/test
```

```
$ PATH=$PATH:`pwd` ; export PATH
```

```
$ somecommand
```

```
Hello world!
```

Login Scripts

- You don't want to enter aliases, set environment variables, set up command line editing, etc. **every time you log in**
- All of these things can be done in a script that is run each time the shell is started
- Location
 - For bash:
 - `~/ .bashrc`
 - For tcsh
 - `~/ .cshrc`
- Whatever you do:
 - `$ cp .bashrc .bashrc.orig #before you do anything with this file!!!!`

Just in case you screw up!!!

`/bin/cp .bashrc.orig .bashrc`

Edit Example .bashrc (partial)

```
alias fisher='ssh fisher.neu.edu'  
alias 'rm'='rm -i'  
alias 'l'='less'  
#Use human-readable filesizes  
alias du="du -h"  
alias df="df -h"  
alias today='date +"%A, %B %-d, %Y"'  
#I'm lazy and don't want to remember, so I made a function!!  
function dict() {  
  
    curl dict://dict.org/d:"$@" ;  
  
}  
PATH=$PATH:/home/cleslin/test; export PATH  
  
echo "Remember Chess... Make it a great day!"
```

Next time you login it will be set, or you can do a:
\$ source .bashrc

Setting up Windows/Linux Dual-Boot

by

Chuck Roesel

Overview

- Backup personal data!
- Have all installation media and keys handy
- Defragment disk
- Shrink Windows partition
- Download Fedora 17 .iso file
- Download and install Fedora USB creator
- Create live USB
- Install Fedora 17 in unused space
- Start using Linux!

Worst-Case Scenario

- You could end up with an unbootable hard drive and have to re-install everything from scratch
- Those recovery disks you created when you took the computer out of the box may not work once partitions are changed
- Without installation keys installation disks are useless
- Be prepared for the worst-case scenario

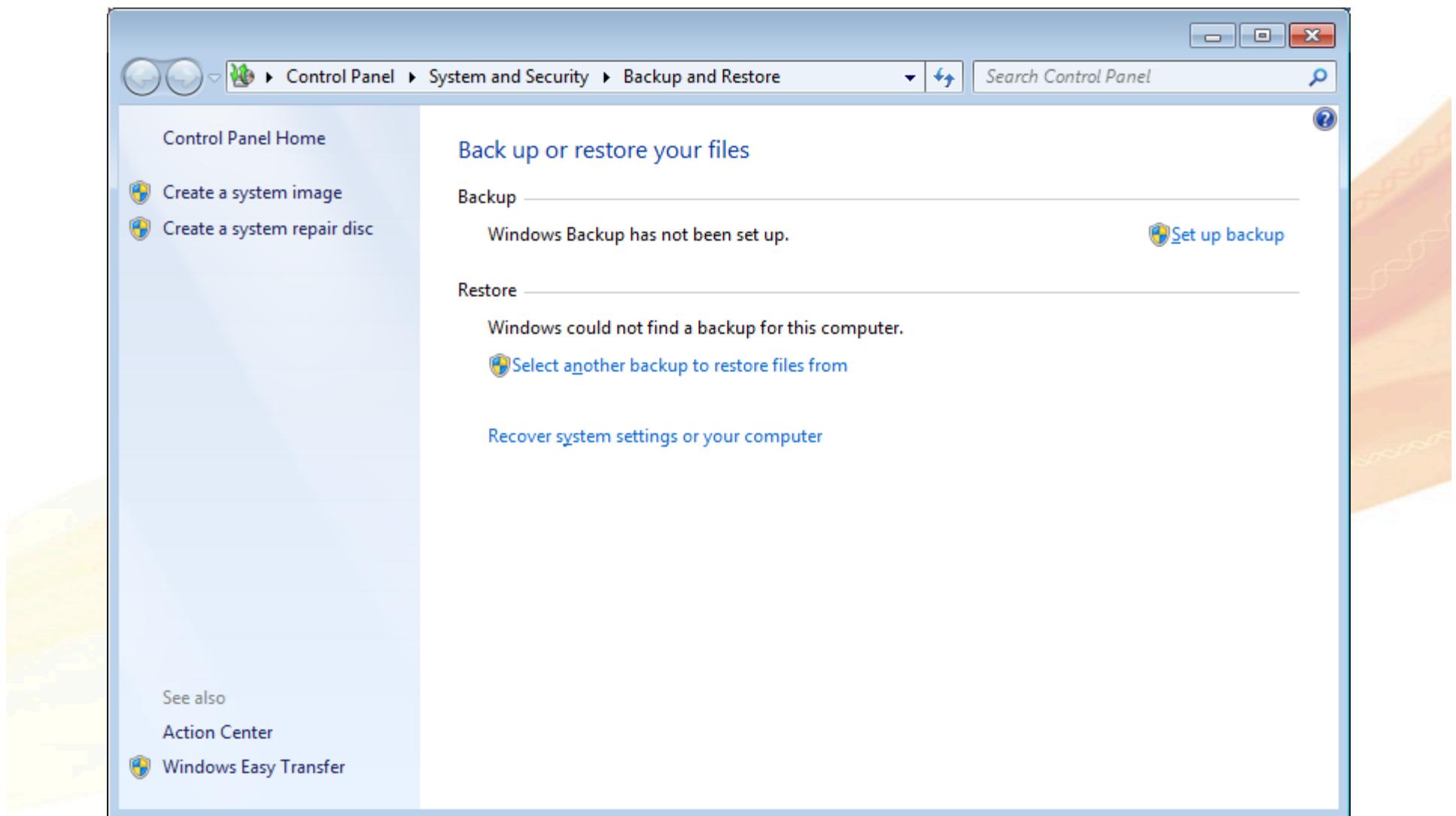
Backup Personal Data Files

- In addition to full system backups, you want a quick and easy way to get to your most critical documents and files
- Copy your personal document and data directories to a USB memory stick or an online repository like DropBox
- You don't want to have to do a full system restore just to get critical files
- DropBox works on Mac OSX, Windows, and Linux

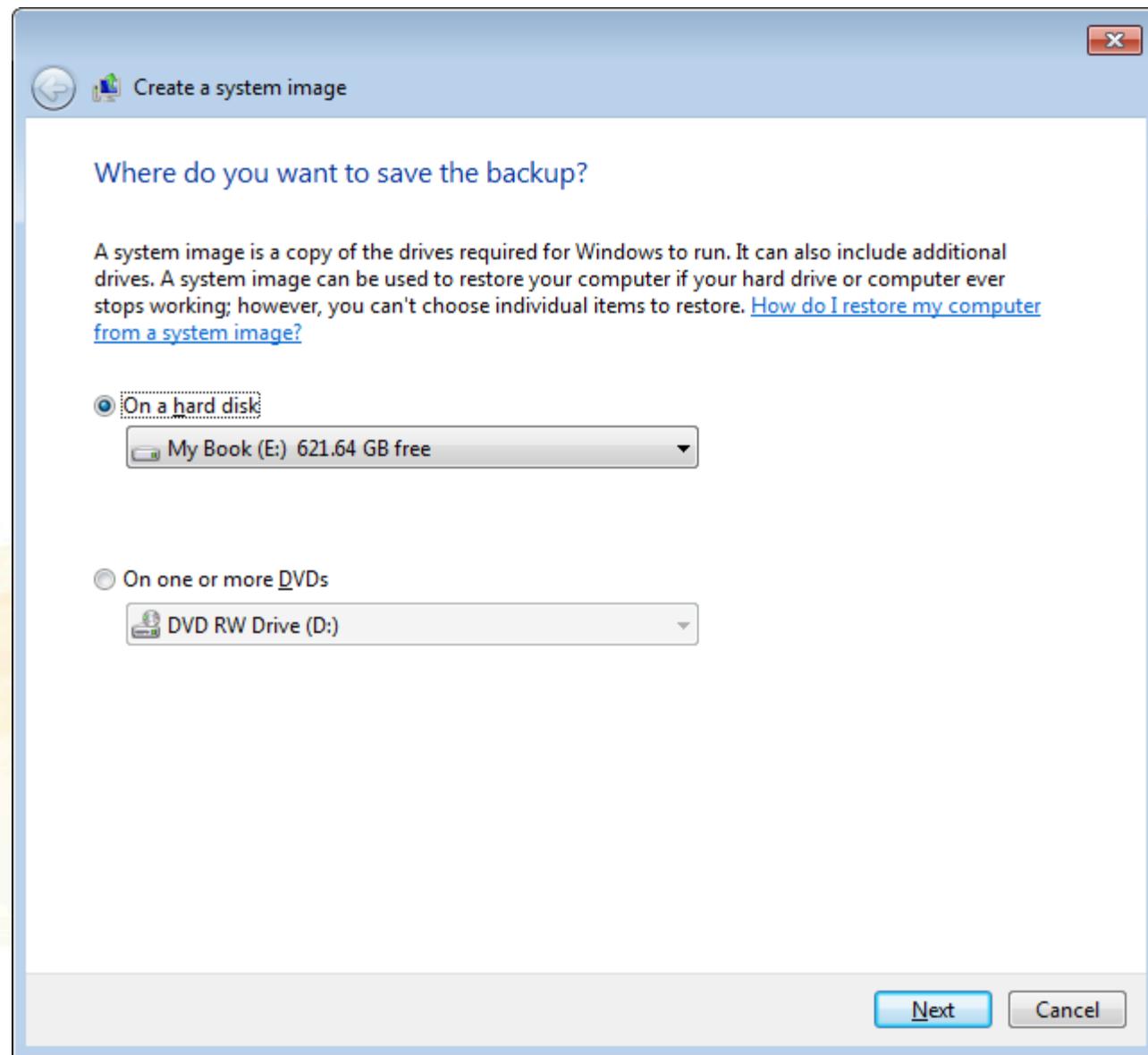
What You will Need



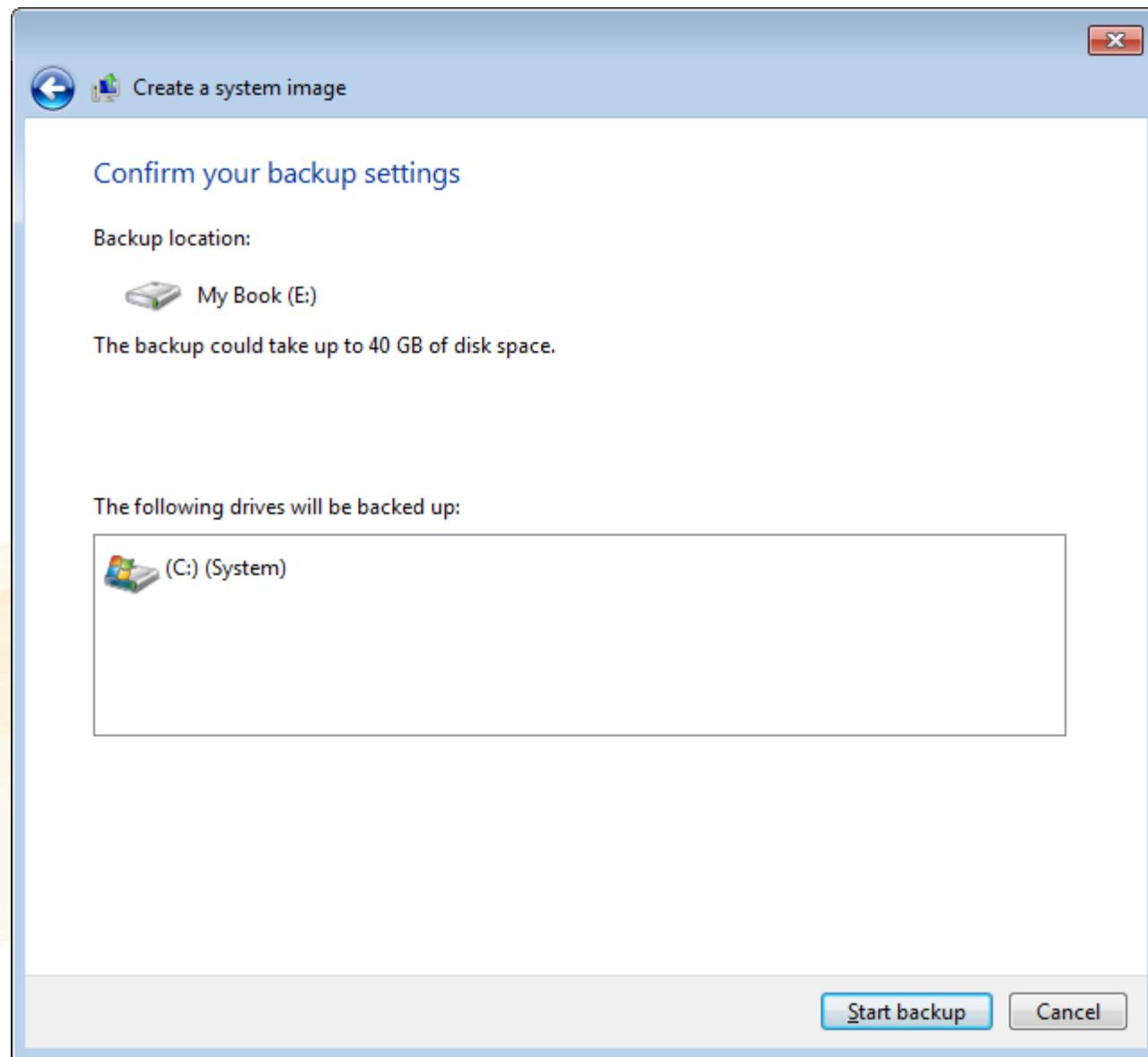
Full System Backup



Create a System Image



Create a System Image



Backup Downloaded Software Installation Media

- Write down all your product installation keys
- Even if you have backup media, without those keys you're out of luck
- Windows keys are usually printed on a label on the bottom of your computer
- If not there, the key may be printed on a certificate included with your PC

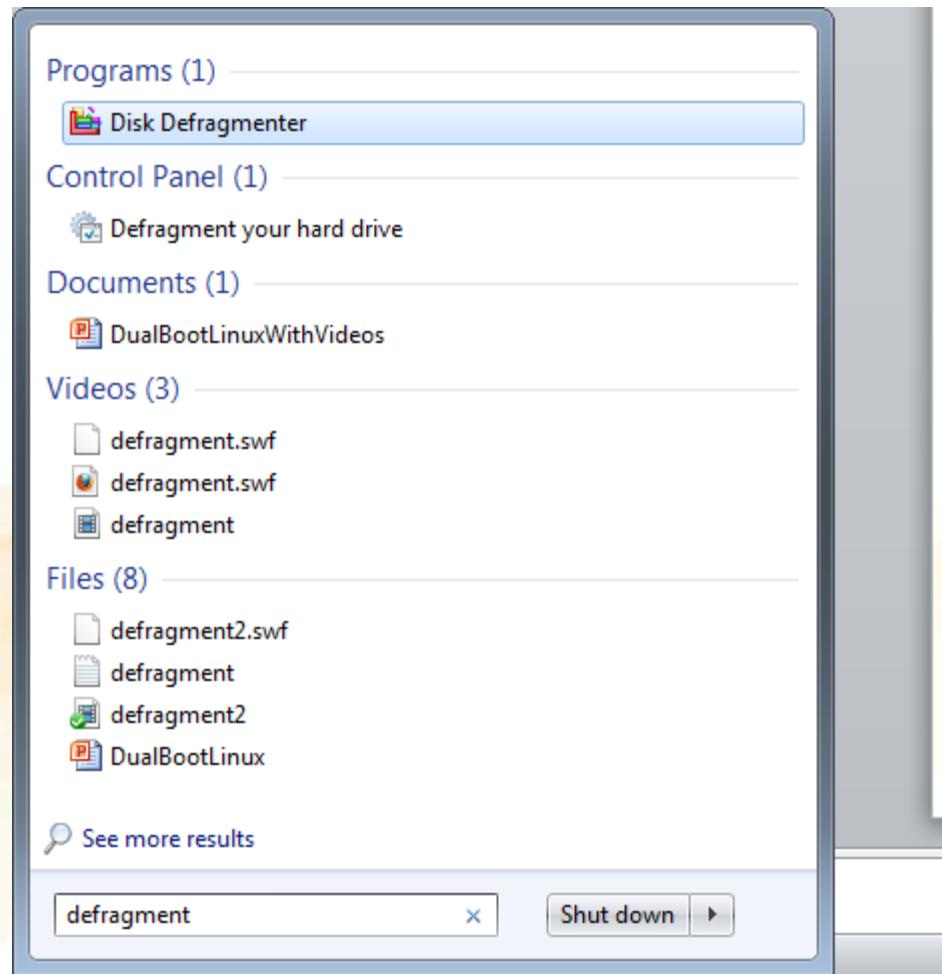
Download Windows 7 Backup Media

- Most PCs ship with Windows pre-installed, but manufacturers don't include backup media
- Those recovery disks they offer as a substitute may not work once partitions are changed
- Fortunately, you can legally download the backup media for free
- See
[www.pcworld.com/article/248995/
how_to_install_windows_7_without_the_disc.html](http://www.pcworld.com/article/248995/how_to_install_windows_7_without_the_disc.html)
- A Google search returns about a million hits for Windows install disks, most of them out to get you. Just go right to PC World link above.

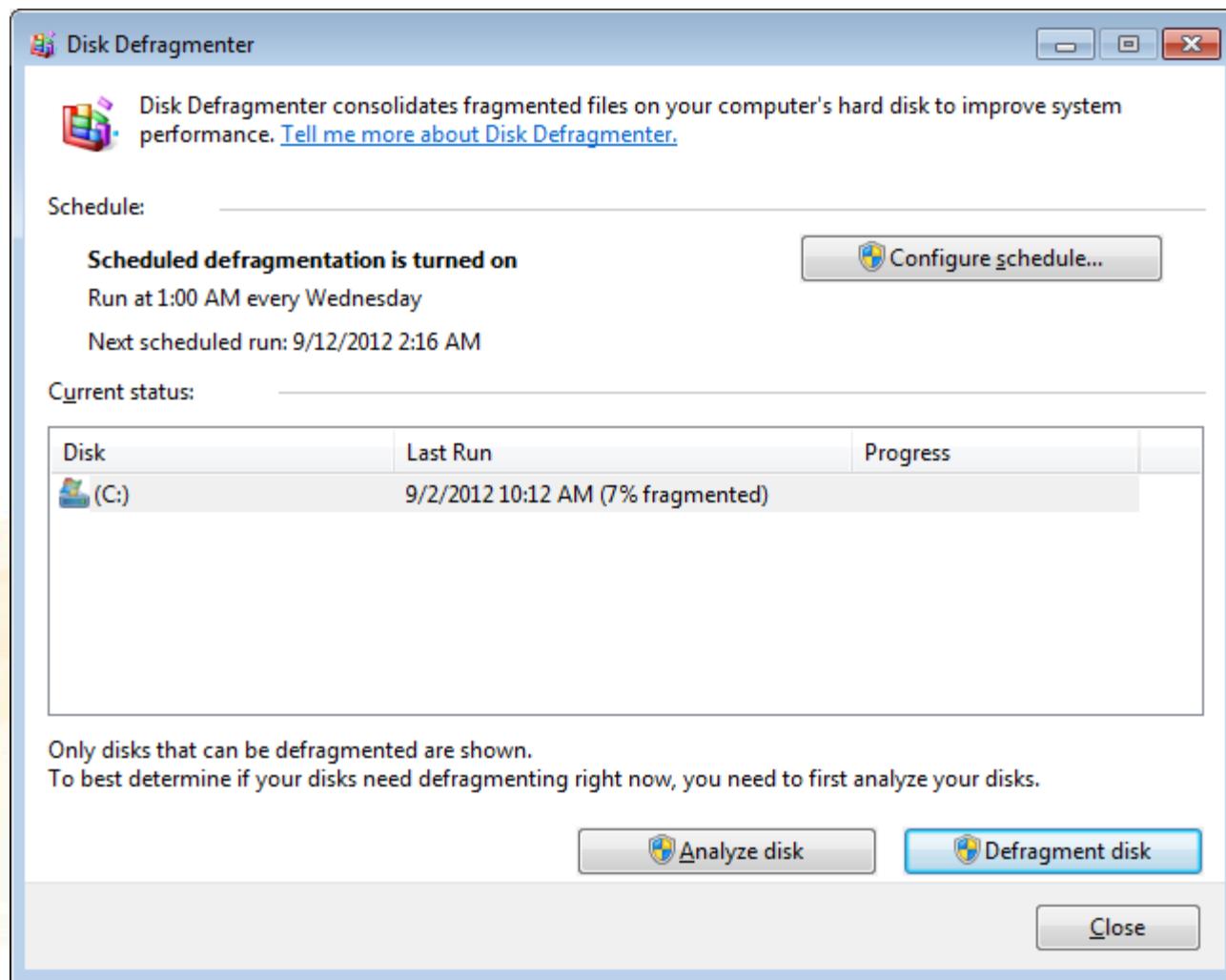
Defragment Disk

- Defragmenting your hard drive consolidates files into a smaller area on your hard drive. This increases the disk space that can be freed for your Linux installation.
- Select
Programs
Accessories
System Tools
Disk Defragmenter
- Depending on the size of your hard drive, number of files, and amount of fragmentation this can take from one to four hours.

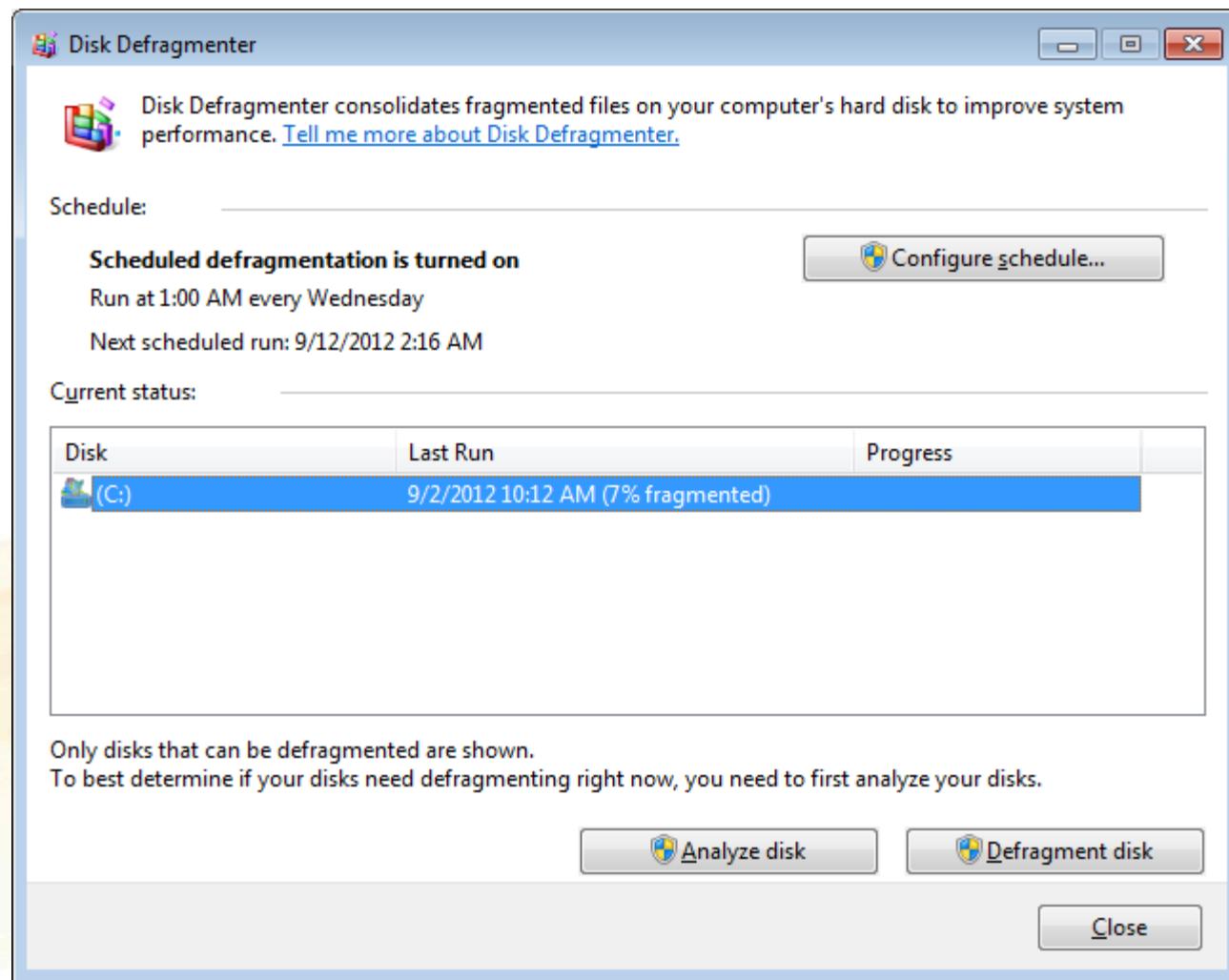
Defragment Disk



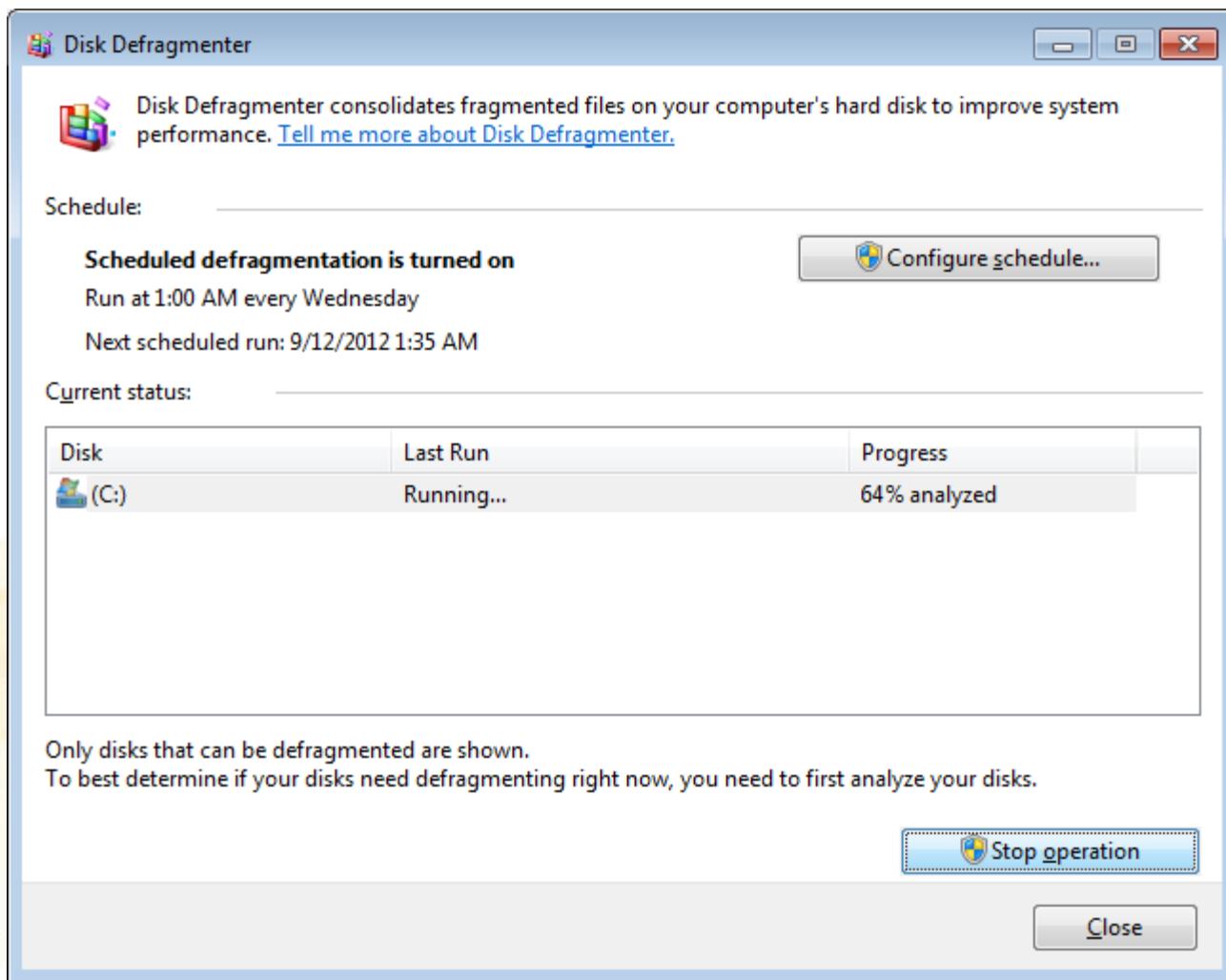
Defragment Disk



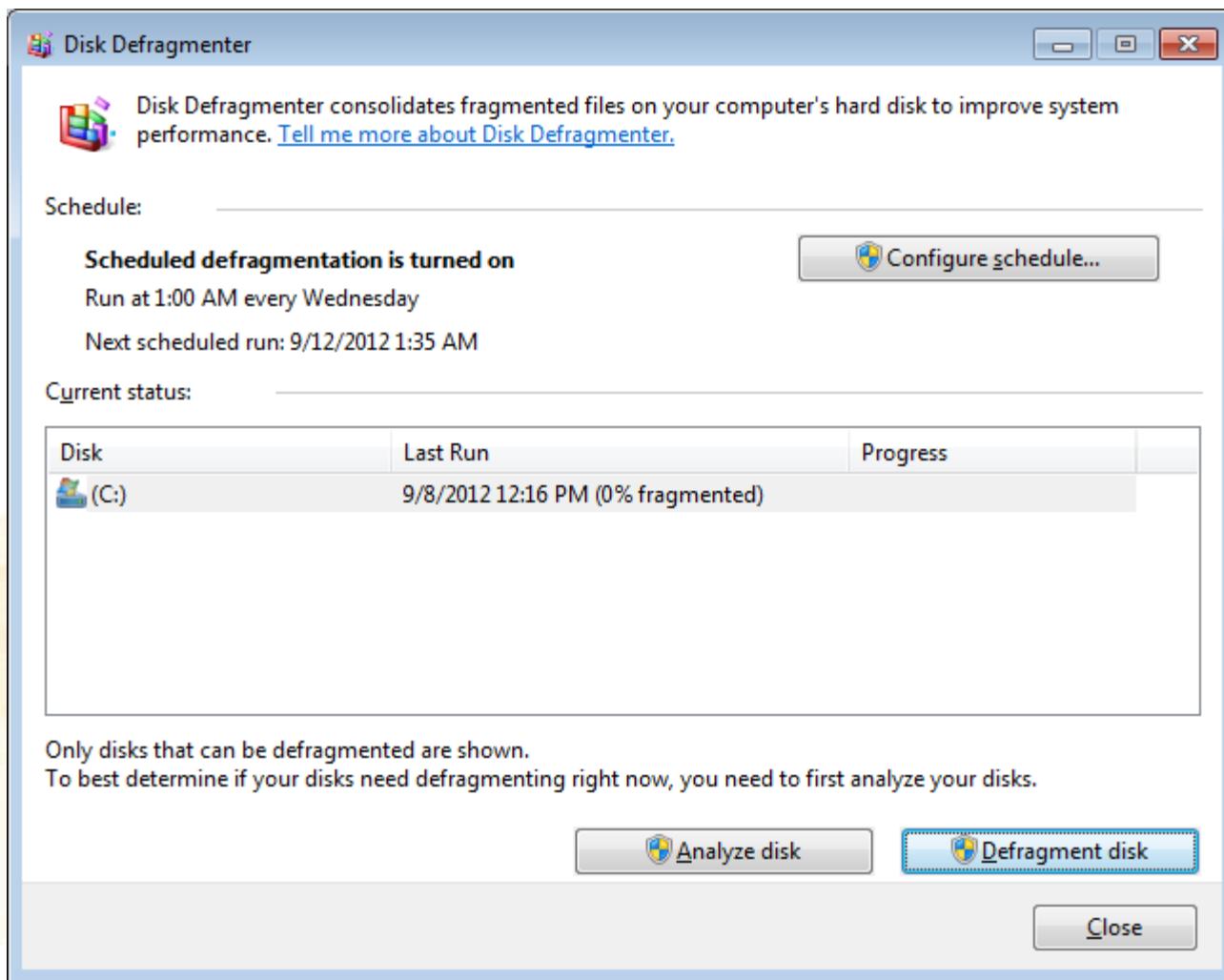
Defragment Disk



Defragment Disk



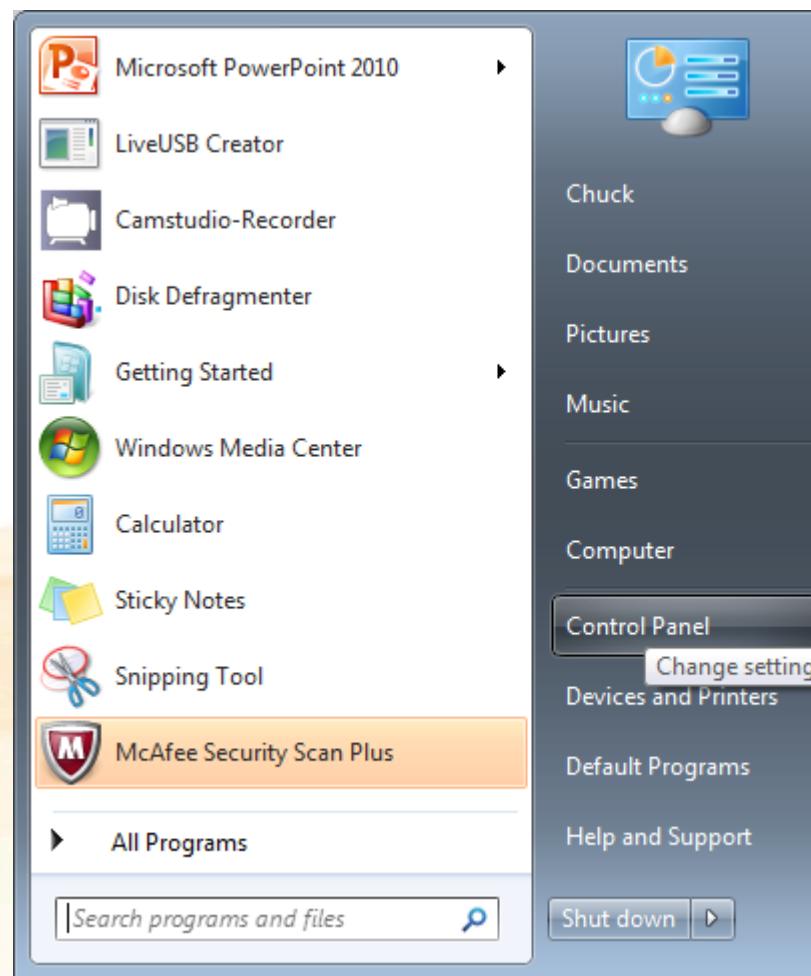
Defragment Disk



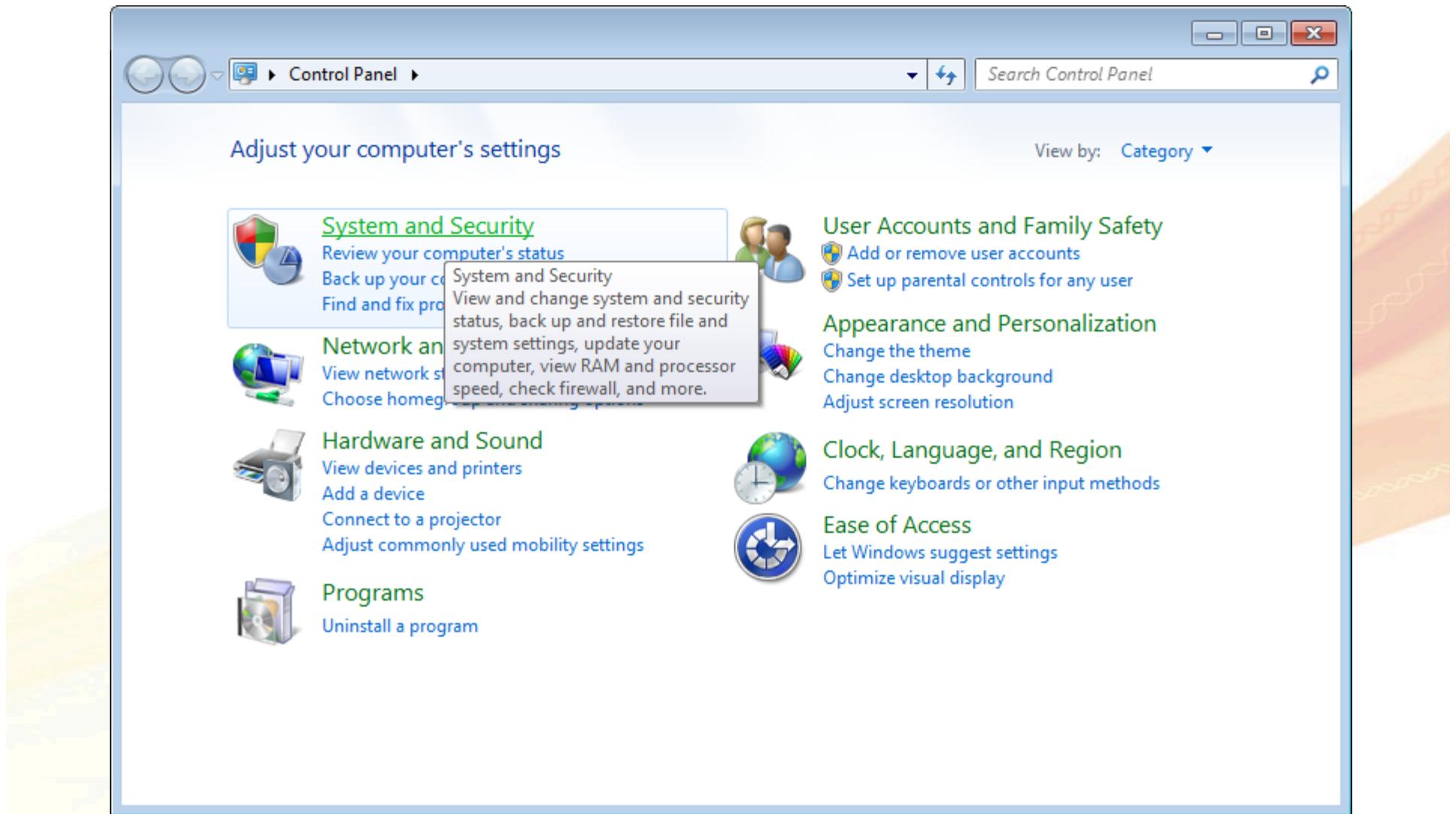
Shrink Partition

- Shrinking your Windows partition will make space available for your Linux installation
- Select
Control Panel
System and Security
Administrative Tools
Create and Format Hard Disk Partitions
- Right-click on your Windows partition and select Shrink Volume.
- Select the new size for the Windows partition. I split my drive into two equal 150G partitions.

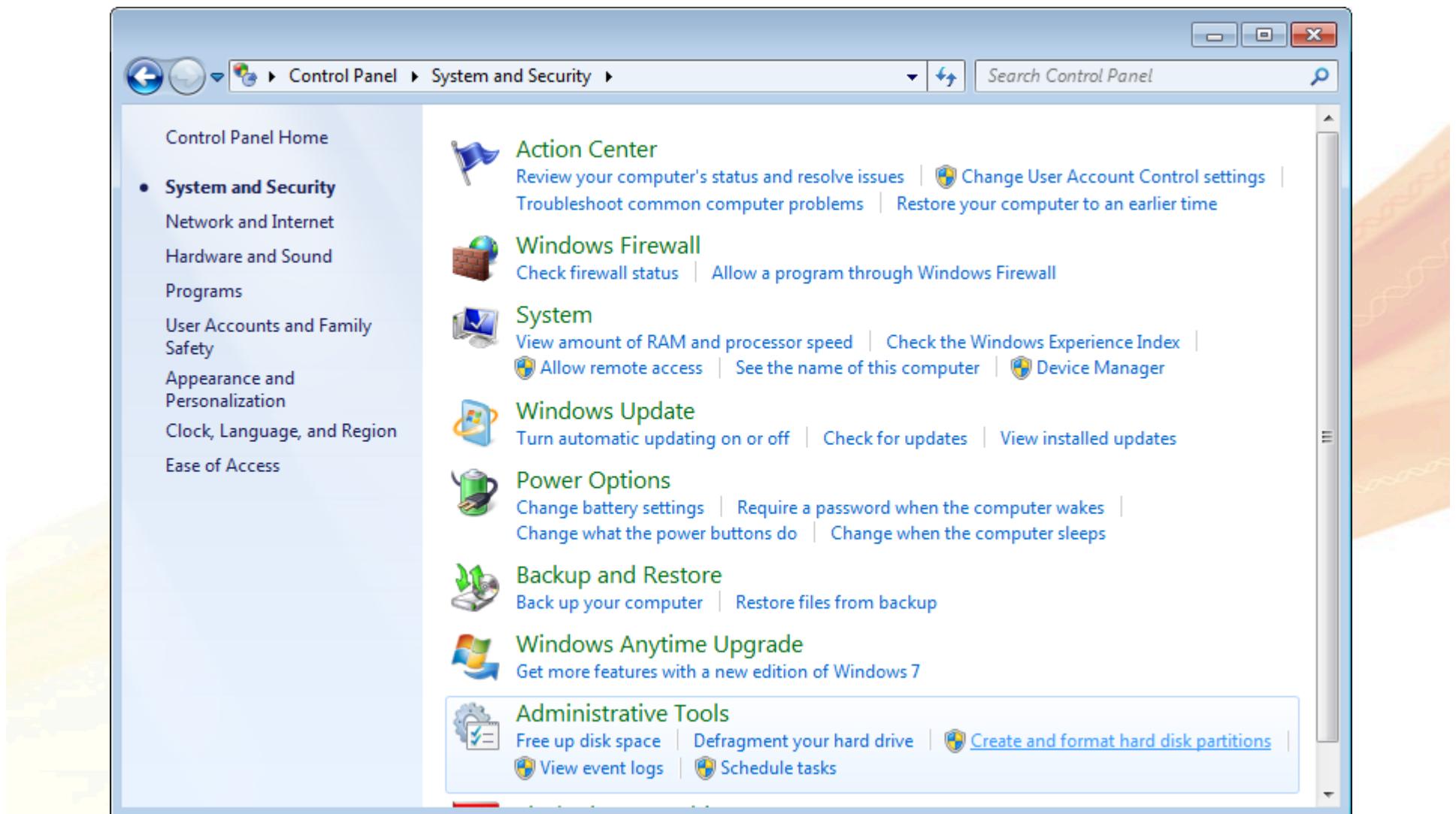
Select Control Panel



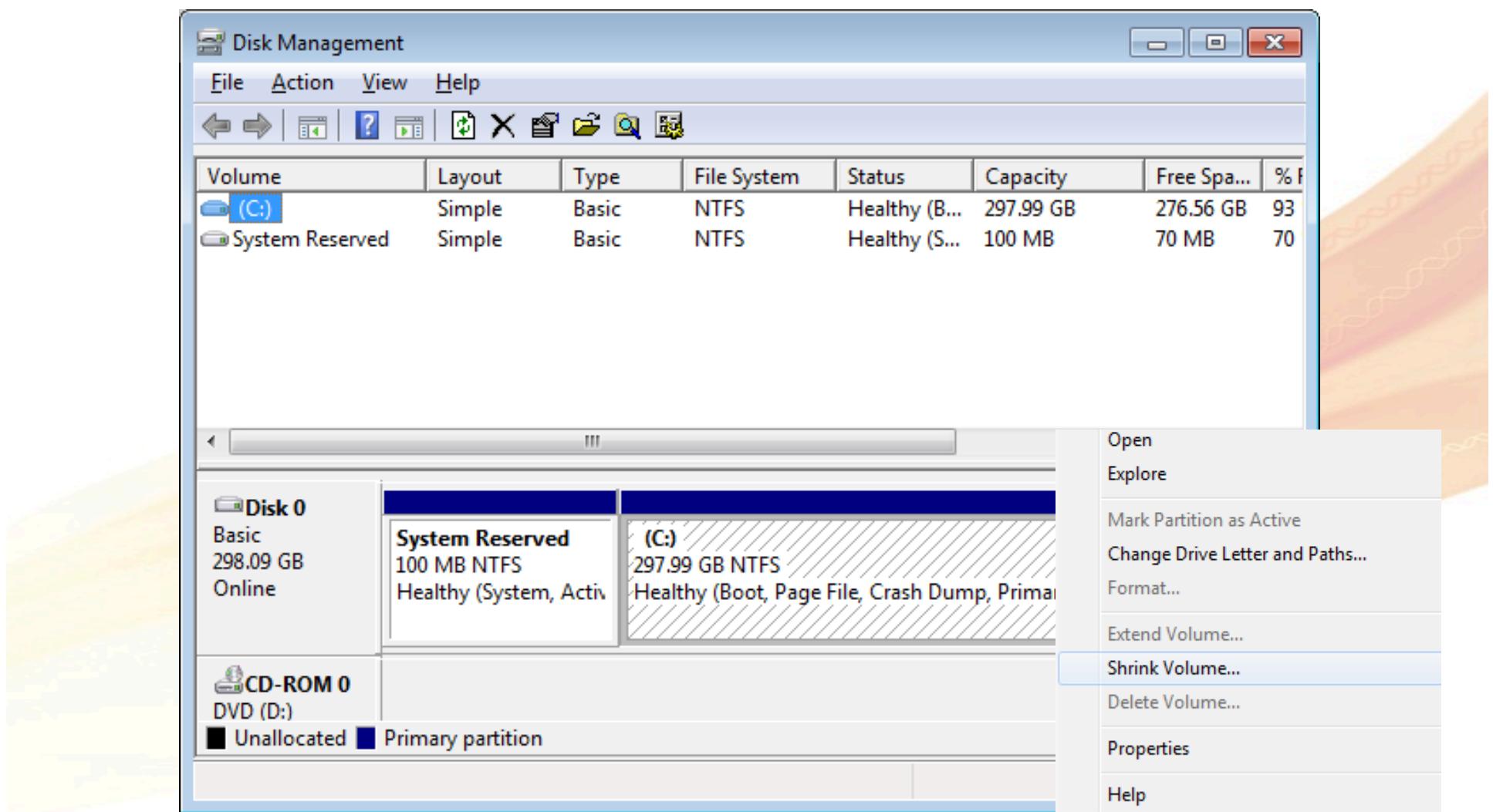
System and Security



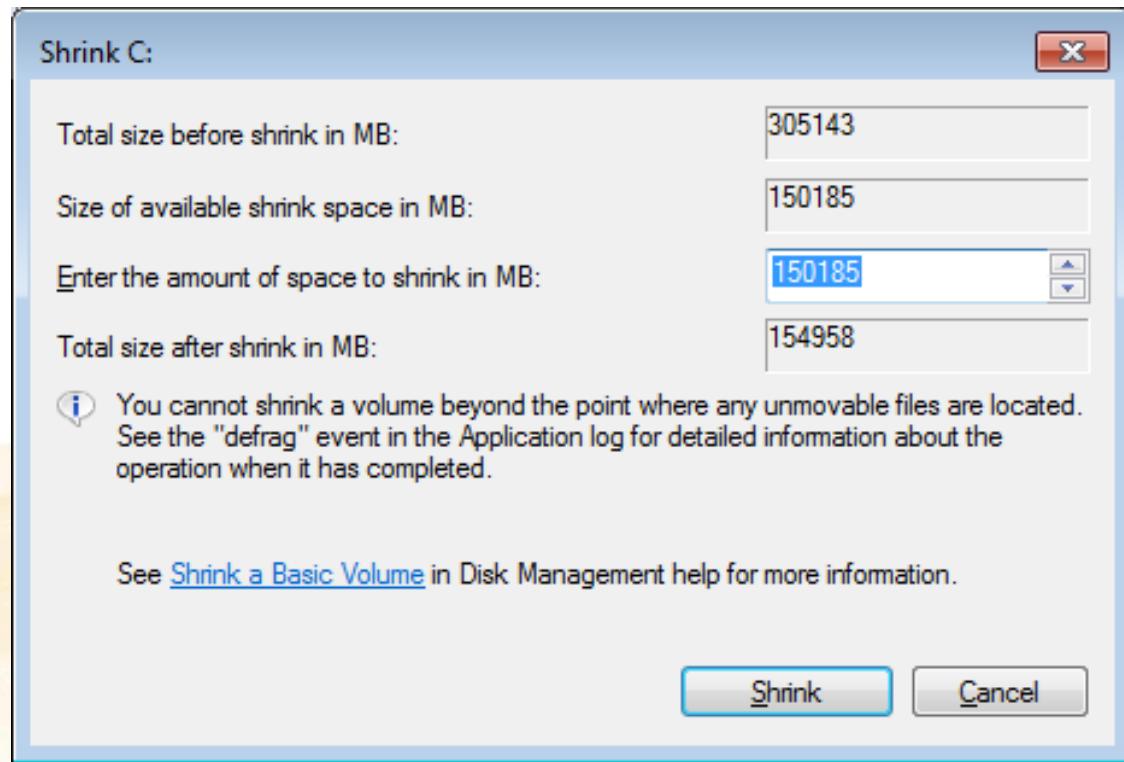
Create and Format Hard Disk Partitions



Right-Click C: and Shrink Volume



Right-Click C: and Shrink Volume



Download Fedora 17

fedoraproject.org/get-fedora

The screenshot shows a web browser window with the URL fedoraproject.org/get-fedora in the address bar. The page itself is the 'Get Fedora' section of the Fedora website. At the top, there's a navigation bar with links for Home, Features & Screenshots, Download, Community, Using Fedora, and Help. Below this is a sub-menu with links for Contributors and a 'Get Fedora' button. A sidebar on the right lists 'WHAT WILL I NEED?' with a bulleted list of system requirements. The main content area features a large image of a DNA helix and text about the free nature of Fedora.

fedoraproject.org/get-fedora

Most Visited Getting Started Latest Headlines Bookmarks Google Bookmark Bookmarks

English OK

A Red Hat-Sponsored Community Project

Home Features & Screenshots Download Community Using Fedora Help

Contributors

Get Fedora

Fedora is 100% free for you to enjoy and share.

FEDORA 17 DESKTOP EDITION

Installable Live Media

This is the latest version of the Fedora Linux operating system's Desktop Edition. It's everything you need to try out Fedora — you don't have to erase anything on your current system to try it out, and it won't put your files at risk. Take Fedora for a test drive, and if you like it, you can install Fedora directly to your hard drive straight from the Live Media desktop if you like.

Download Now!

645MB, ISO format image for Intel-compatible PCs (64-bit)

WHAT WILL I NEED?

- ✓ A blank CD or DVD **or** a blank 1GB+ USB stick.
- ✓ A 400MHz or faster processor
- ✓ At least 768 MB memory (RAM), 1 GB recommended for best performance
- ✓ At least 10 GB hard drive space (only required for

Download Fedora 17

fedoraproject.org/en/get-fedora-options

The screenshot shows a Firefox browser window displaying the fedoraproject.org/en/get-fedora-options page. The page has a green header with the title "Get Fedora" and a sub-header "Fedora is 100% free for you to enjoy and share." Below the header, there are tabs for "Desktops", "Formats", "Spins", and "Clouds". The "Desktops" tab is selected.

FEDORA DESKTOPS

There are multiple desktops available for use with Fedora. Each has a slightly different look and feel and offers varying levels of customization. Desktops affect how windows, icons, menus, and various basic desktop interfaces look and feel. You may want to try a few different ones to see which one is best for you!

Fedora 17 Desktop Edition (Fedora Default)

The GNOME-based default Fedora Desktop

- [Fedora Desktop spin homepage >](#)
- [GNOME.org website >](#)

32-bit [Download Now!](#)
646MB CD ISO

64-bit [Download Now!](#)
645MB CD ISO

A large blue arrow points diagonally across the page from the left towards the "64-bit" download link, with the text "64 bit for most newer PCs" written along its path.

HELP, I JUST WANT FEDORA!
There are a lot of options on this page. Would you rather download the standard Fedora image?

[Get it here.](#)

HANDY RESOURCES

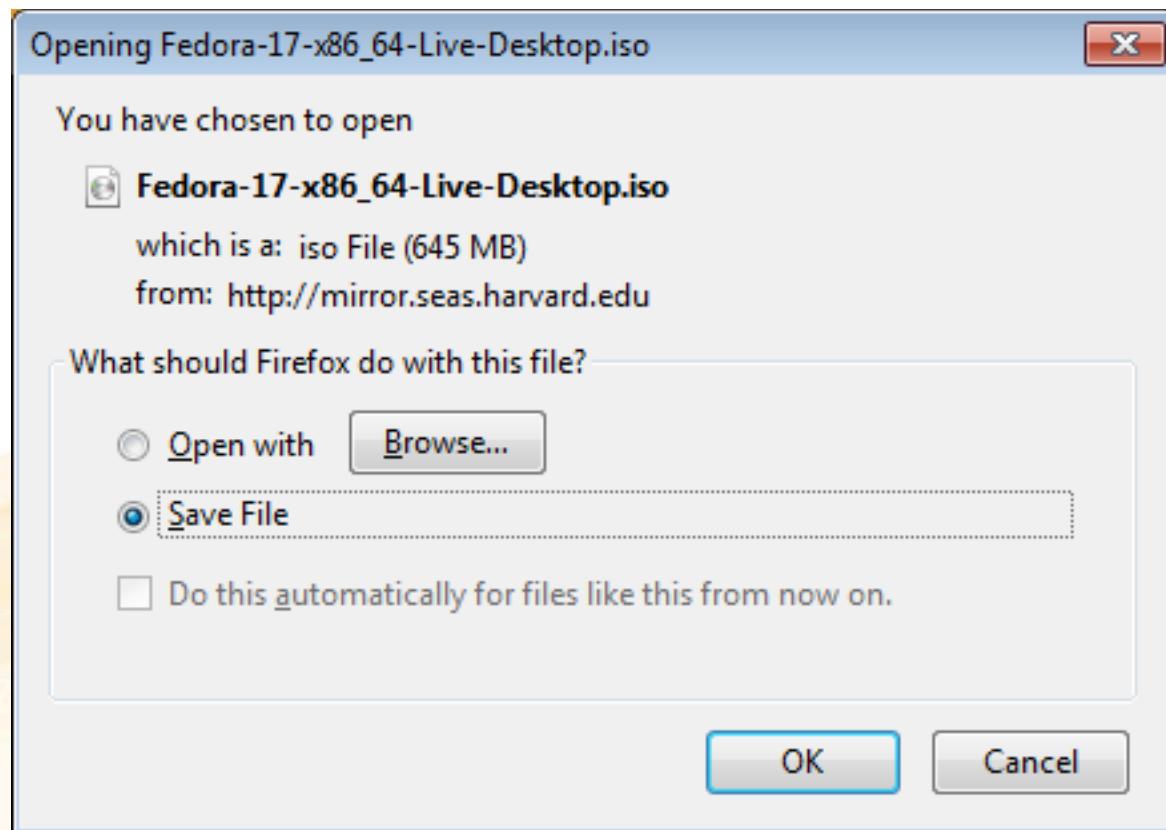
- [Installation Guide](#)
- [Release Notes](#)
- [Common Bugs](#)
- [Upgrading Fedora](#)
- [Help & Support](#)
- [Verify Download](#)

THERE'S MORE FEDORA!

fedora SPINS

OTHER WAYS TO GET FEDORA
We offer several alternative

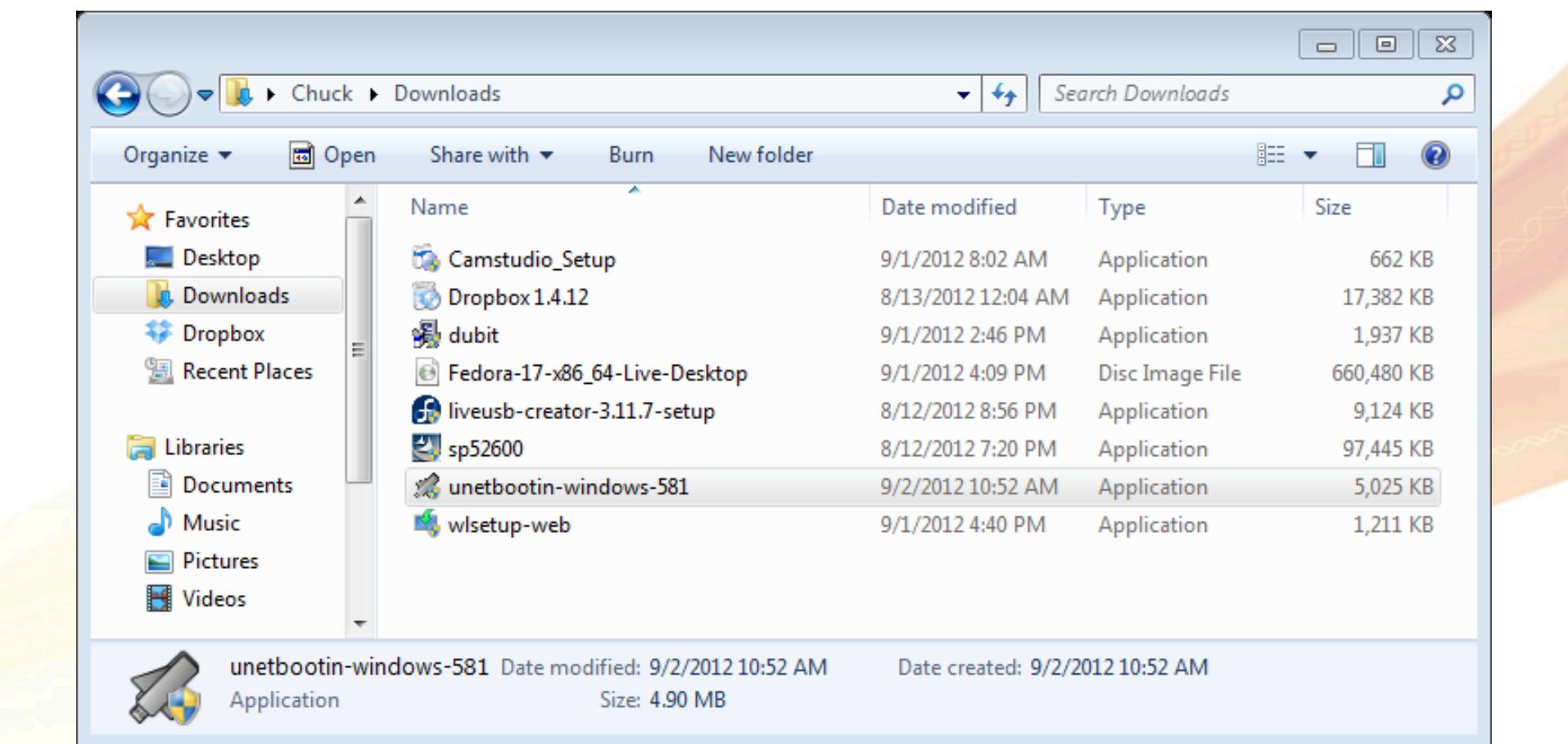
Download Fedora 17



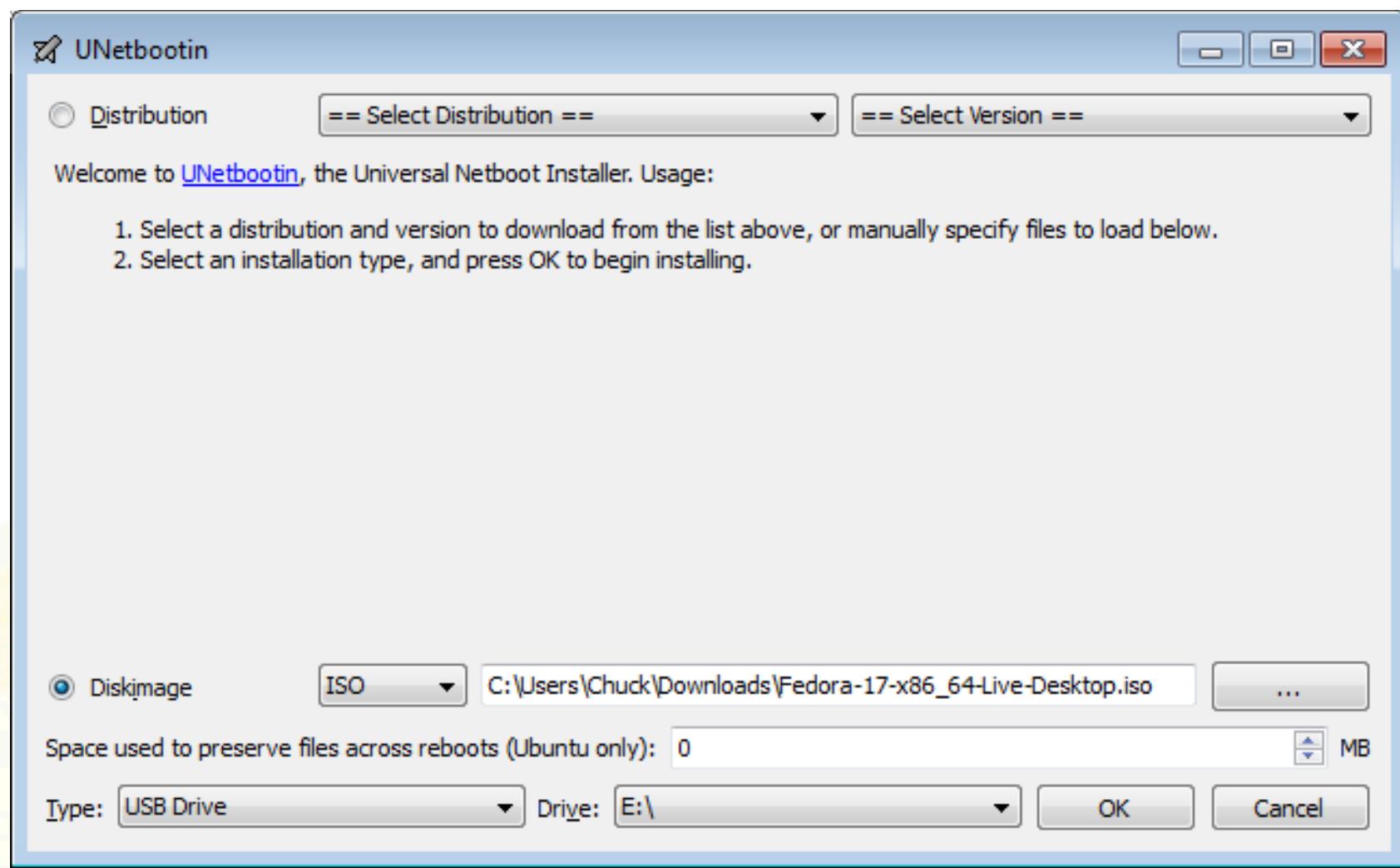
Download Unetbootin

The screenshot shows a Firefox browser window with the title bar "Firefox" and the address bar "UNetbootin - Homepage and Downloads". The main content area displays the UNetbootin homepage. At the top, there are three large blue buttons with icons: a Windows logo for "Download (for Windows)", a Linux penguin icon for "Download (for Linux)", and a Mac OS X icon for "Download (for Mac OS X)". Below these buttons, the text "Packages: [Ubuntu](#) [Debian](#) [Fedora](#) [Suse](#) [Arch](#) [Gentoo](#) [More](#)" is listed. The "More" link is underlined. The page content includes sections for "Introduction", "Requirements", and "Features". The "Introduction" section explains what UNetbootin does and how it works. The "Requirements" section lists the operating systems and internet access needed. The "Features" section describes the software's capabilities. On the right side of the page, there is a sidebar with a list of links: "» [Introduction](#)", "» [Requirements](#)", "» [Features](#)", "» [Installation & Screenshots](#)", "» [Removal Instructions](#)", "» [Other Distributions](#)", "» [Translations](#)", "» [FAQs](#)", "» [License & Credits](#)", and "» [UNetbootin Wiki](#)".

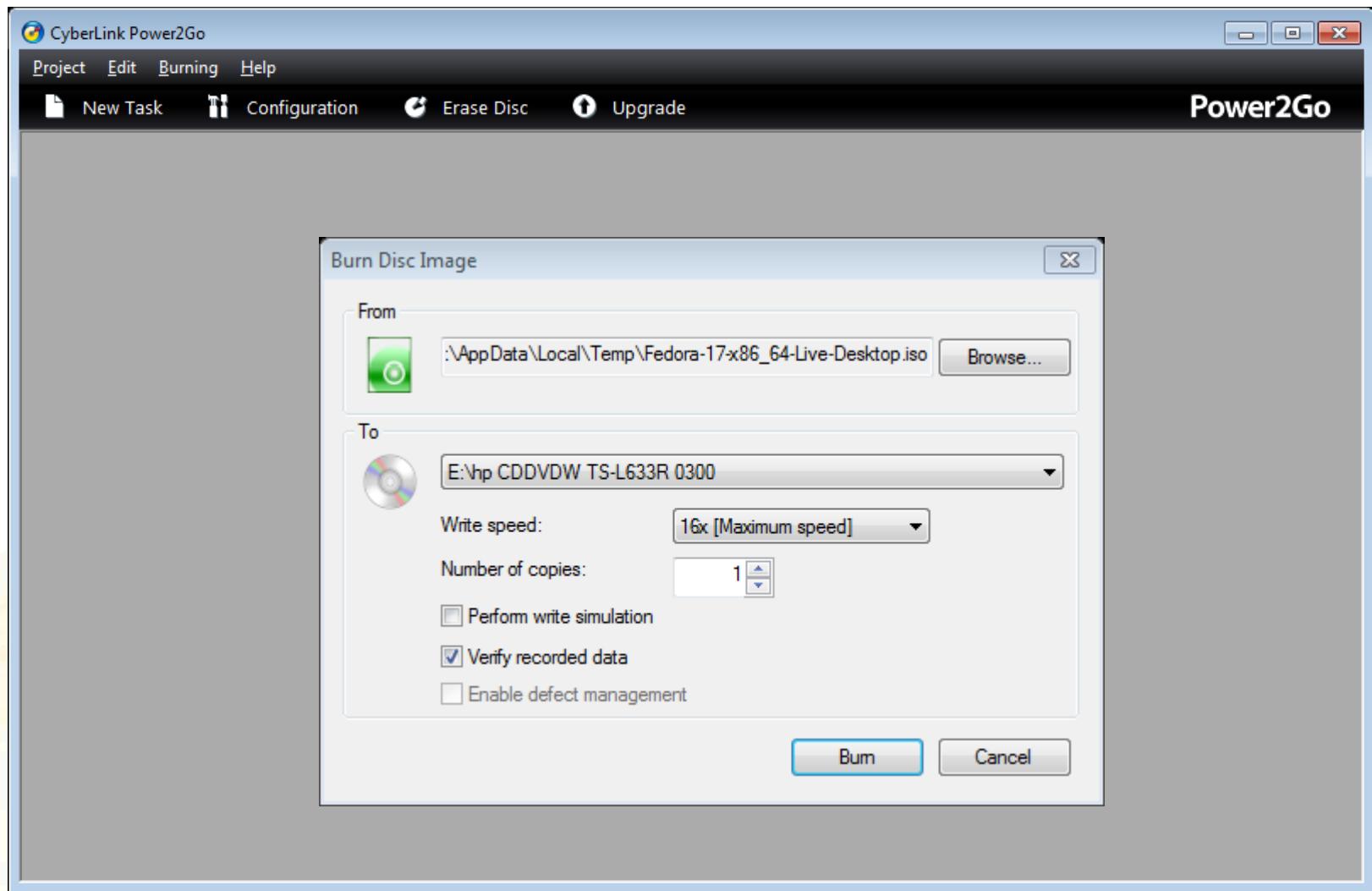
Double-Click Unetbootin-windows-581



Create Live USB



Or Burn Disc Image if Your PC Can't Boot from USB



Configure System to Boot from USB

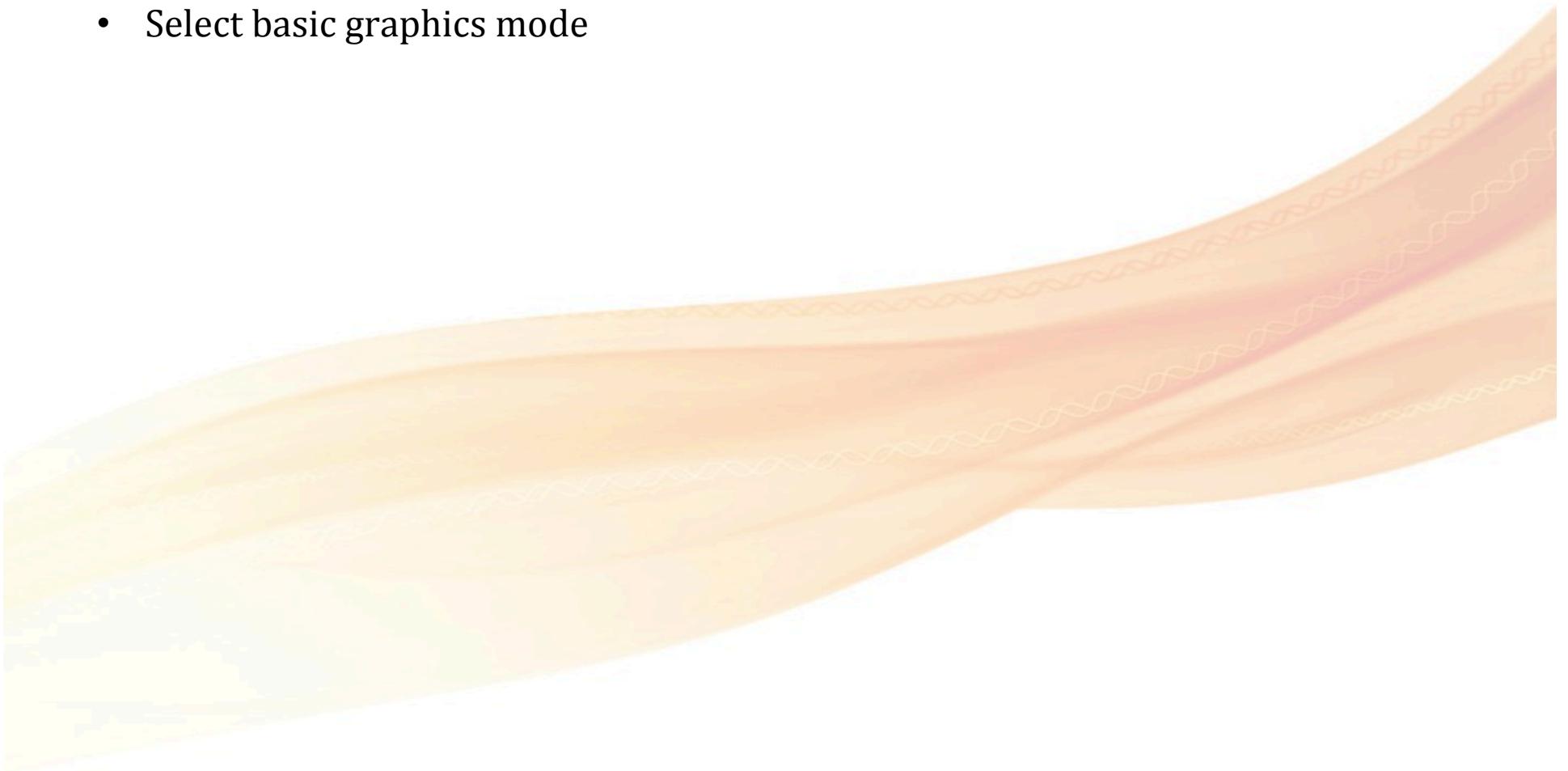
- At power on, press ESC, F2, or ThinkVantage button (varies by manufacturer) to enter system setup. Usually text in the lower left corner of screen indicates how to do this.
- Press function key to enter BIOS setup (F10 on HP Pavilion). This also varies by manufacturer, so look for a menu with instructions for updating BIOS.
- Look for a System Configuration tab or something similar
- Select Boot Options and press enter
- Select Boot Order and press enter
- Arrow down to USB Diskette on Key/USB Hard Disk
- Press F6 until this is the first device listed under boot devices.
- Press F10 to save and exit

BIOS Main Tab

InsydeH2O Setup Utility		Rev. 3.5			
Main	Security	Diagnostics	System Configuration	Exit	
System Time	[12: 16: 50]				
System Date	[06/23/2009]	Item Specific Help			
NoteBook Model	HP Pavilion dv6 Notebook PC	<Tab>, <Shift-Tab>, or <Enter> selects field.			
Product Number	UT1SKU7#ABA				
System Board ID	3060				
Processor Type	AMD Turion(tm) x2 Dual-Core Mobile RM-74				
Processor Speed	2200 MHz				
Total Memory	2048 MB	Video memory is included in the total memory and the size is dynamically changed in the OS based on the usage.			
BIOS Version	F.01				
Serial Number	[REDACTED]				
UUID Number	434E4639-3031-3035-365A-00238B51FFD				
Product configuration ID 01100000140020000					
Factory installed OS	Vista				
F1 Help	↑ ↓ Select Item	F5/F6 Change Values	F9 Setup Defaults		
Esc Exit	← → Select Menu	Enter Select ▶ SubMenu	F10 Save and Exit		

Insert USB Stick and Reboot

- Select troubleshooting options
- Select basic graphics mode



Install to Hard Drive



Select Keyboard

Fedora Installer

Select the appropriate keyboard for the system.

- Romanian Standard
- Romanian Standard Cedilla
- Russian
- Serbian
- Serbian (latin)
- Slovak (qwerty)
- Slovenian
- Spanish
- Swedish
- Swiss French
- Swiss French (latin1)
- Swiss German
- Swiss German (latin1)
- Tajik
- Tamil (Inscript)
- Tamil (Typewriter)
- Turkish
- Ukrainian
- United Kingdom
- U.S. English**
- U.S. International

Back Next

Select Basic Storage Devices

Fedora Installer X

What type of devices will your installation involve?

Basic Storage Devices

Installs or upgrades to typical types of storage devices. If you're not sure which option is right for you, this is probably it.

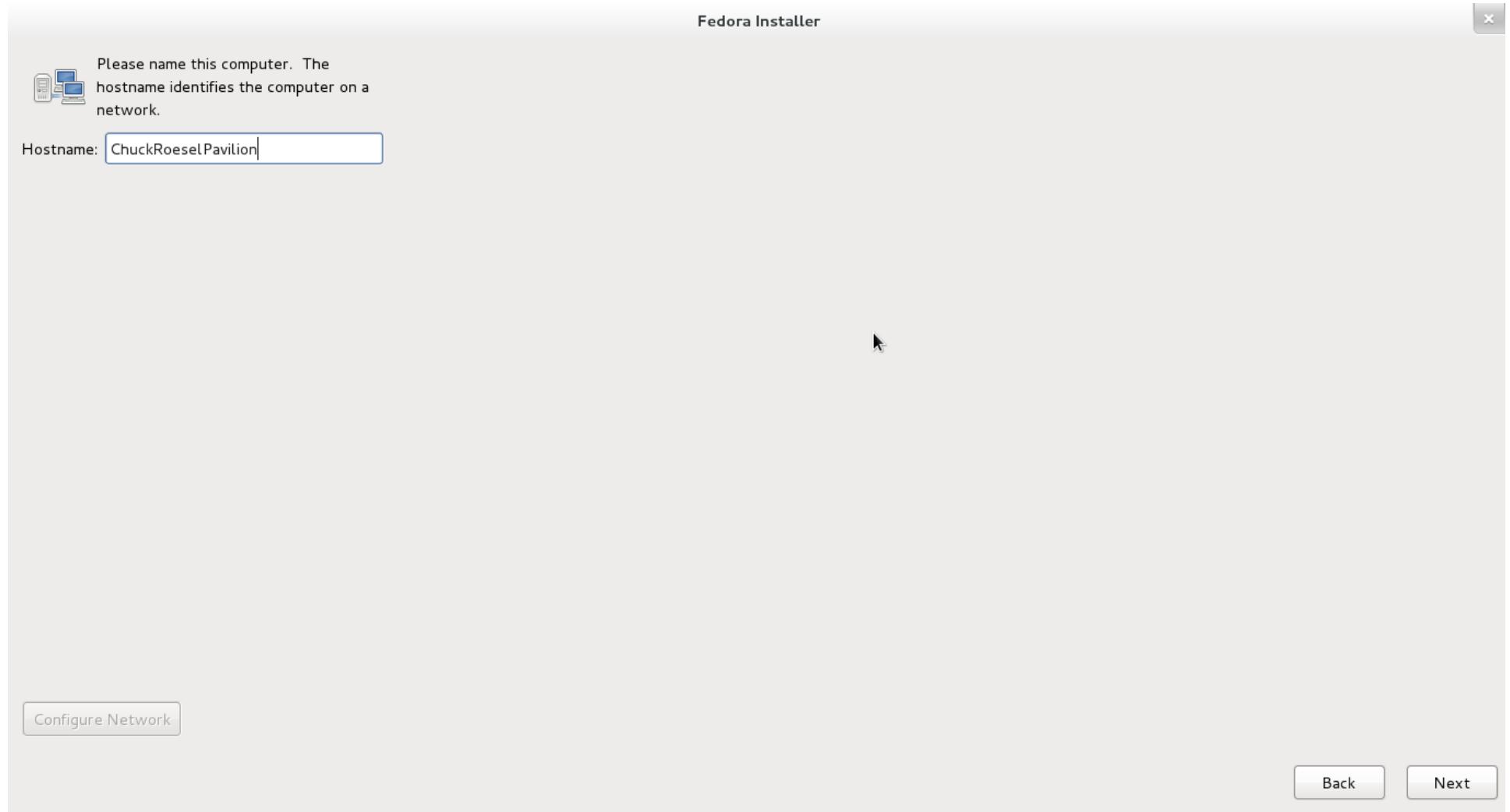
Specialized Storage Devices

Installs or upgrades to enterprise devices such as Storage Area Networks (SANs). This option will allow you to add FCoE / iSCSI / zFCP disks and to filter out devices the installer should ignore.



Back Next

Enter Hostname



Create Root Password

Fedora Installer X

 The root account is used for administering the system. Enter a password for the root user.

Root Password:

Confirm:

Back Next

Select Use Free Space

Fedora Installer X

Which type of installation would you like?

Use All Space
Removes all partitions on the selected device(s). This includes partitions created by other operating systems.

Tip: This option will remove data from the selected device(s). Make sure you have backups.

Replace Existing Linux System(s)
Removes all Linux partitions on the selected device(s). This does not remove other partitions you may have on your storage device(s) (such as VFAT or FAT32).

Tip: This option will remove data from the selected device(s). Make sure you have backups.

Shrink Current System
Shrinks existing partitions to create free space for the default layout.

Use Free Space
Retains your current data and partitions and uses only the unpartitioned space on the selected device(s), assuming you have enough free space available.

Create Custom Layout
Manually create your own custom layout on the selected device(s) using our partitioning tool.

Use LVM
 Encrypt system
 Review and modify partitioning layout

Back Next

Select Hard Drive as Install Target

Fedora Installer

Below are the storage devices you've selected to be a part of this installation. Please indicate using the arrows below which devices you'd like to use as data devices (these will not be formatted, only mounted) and which devices you'd like to use as installation target devices (these may be formatted). Please also indicate which system drive will have the bootloader installed.

Data Storage Devices (to be mounted only)

Model	Capacity	Vendor	Identifier
PNY USB 2.0 FD	7728 MB	PNY	pci-0000:00:1d.0-usb-0:1.1:1.0-scsi-0:0:0:0

Install Target Devices

Boot Loader	Model	Capacity	Identifier
<input checked="" type="radio"/>	ATA Hitachi HTS54503	305245 MB	sda

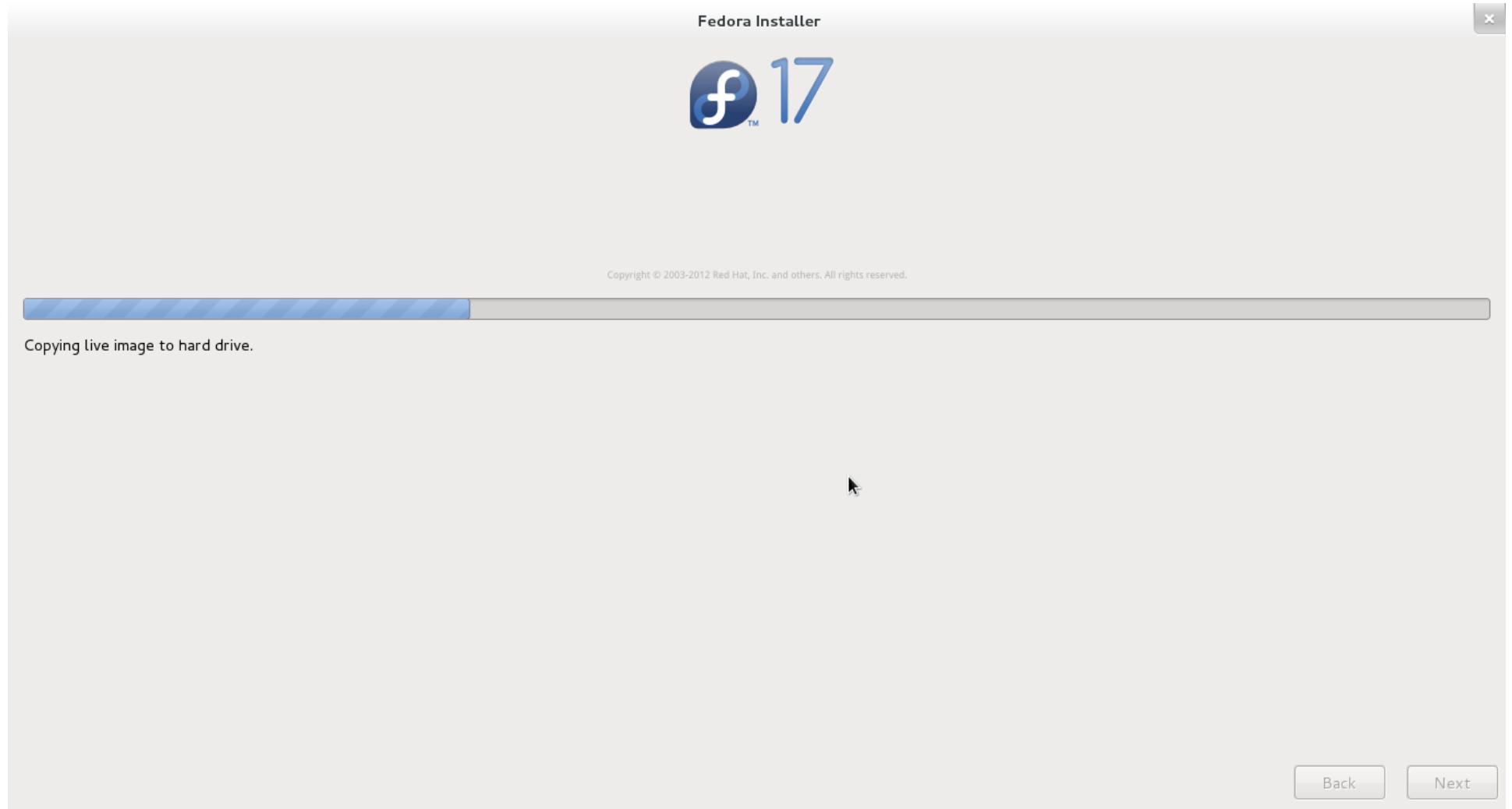
 **Tip:** Your filesystems on the install target devices will not be reformatted unless you choose to do so during customization.

Back Next

Confirm Partition Changes



Image Copy Begins



Reboot

Fedora Installer



Congratulations, your Fedora installation is complete.

Please reboot to use the installed system. Note that updates may be available to ensure the proper functioning of your system and installation of these updates is recommended after the reboot.



Back

Reboot

Done!



Things To Do For Thursday

- Review material
- Understand the commands
 - Log on to fisher (or local machine) and try these things out!!
 - Practice, Practice, and more Practice
- Setup a dual boot, if you want one