

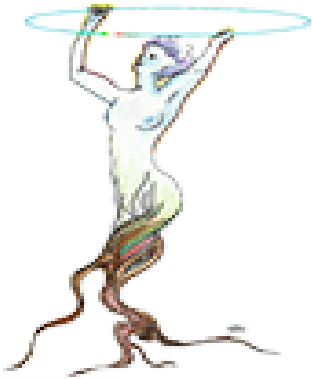
Using Linux in the NSL

Mark Suhovecky
August 17th, 2010

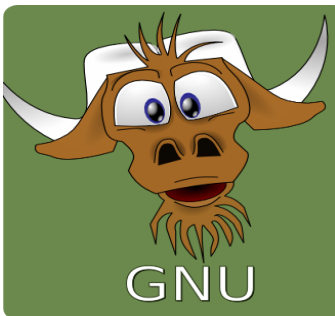
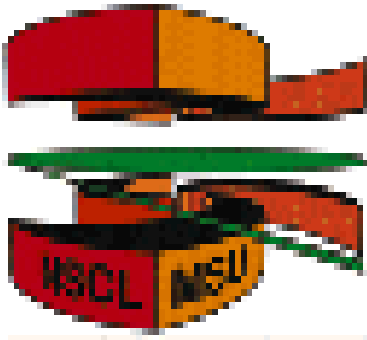
Outline

- Linux in the NSL, Connecting to NSL Machines
- I'm in Linux- now what? The Basics
- Uploading, Downloading & Archives
- Living with AFS
- Modules
- Other Helpful Linux Things

Linux Use in the NSL



Geant 4



- Huge amount of analysis software written to run on Linux.
- Our data acquisition systems are Linux machines (MSU, ORNL)
- Free Open Source Development Tools (GNU, gcc, QT).

General-Use NSL Linux Machines

- **alpher.phys.nd.edu** (ND AFS, 32-bit)
- **arcturus.phys.nd.edu** (ND AFS, 64-bit)
- **regulus.phys.nd.edu** (CRC AFS, 64-bit)
- **zozma.phys.nd.edu** (ND AFS, 32-bit)
- 3 Linux desktop machines on NSL mezzanine (ND AFS, 32-bit)

NSL Linux Data Acquisition Machines

MSU NSCL Systems: (Used by **ams**, **ast**, **nuc**, **react** groups)

- **bethe.phys.nd.edu**
- **fermi.phys.nd.edu**

ORNL Systems: (Used by **rnb** group)

- **ampere.phys.nd.edu**
- **coulomb.phys.nd.edu**

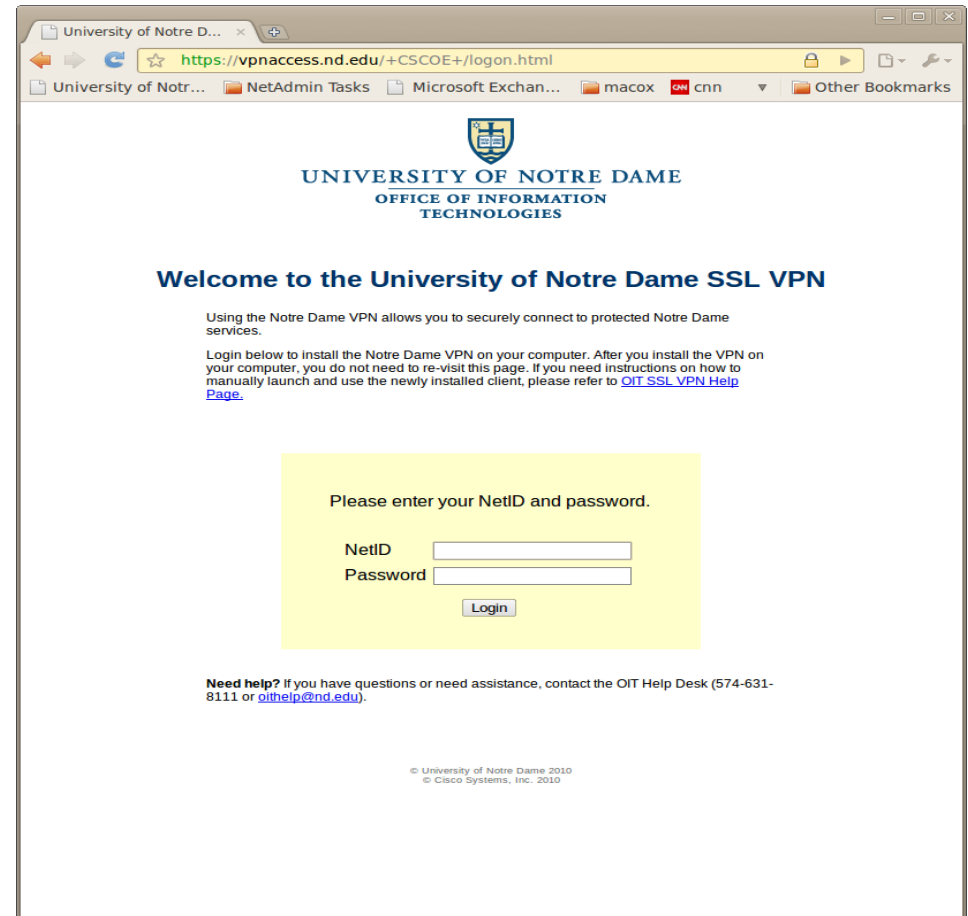
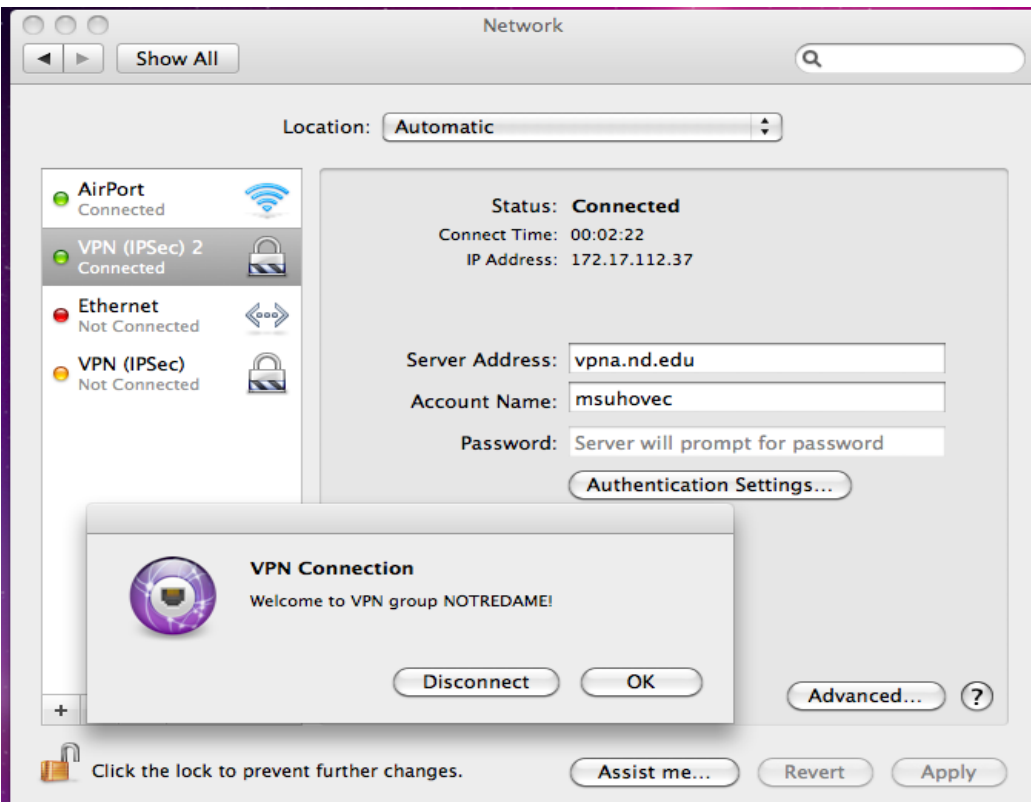
FN Console Dual Display (Display Driver, image capture, et. al.)

- **fowler2.phys.nd.edu**

Connecting to NSL Machines

Off Campus? Use VPN

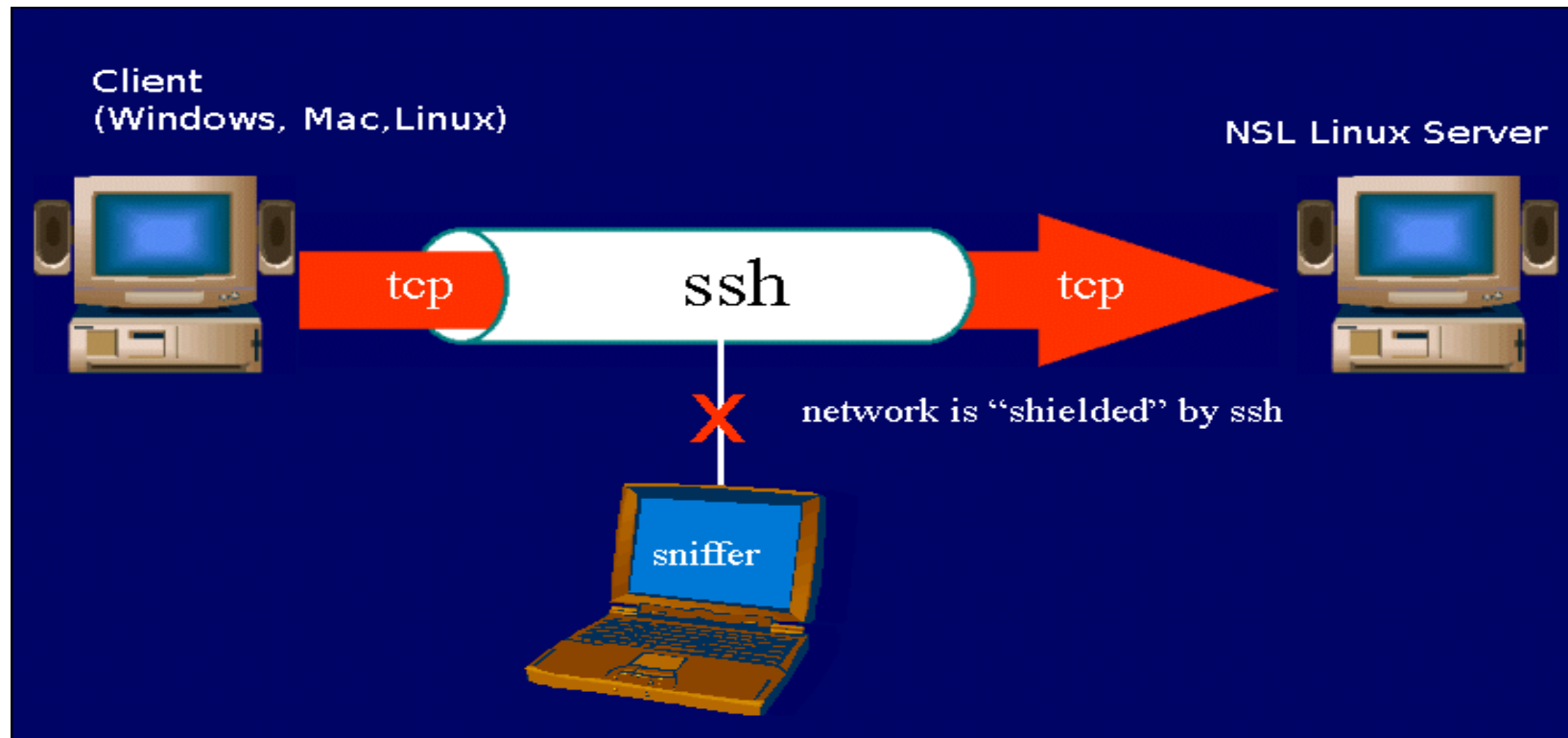
For off-campus connections, use a Virtual Private Network (VPN) when you connect.



Connecting to NSL Machines Prerequisite 1: SSH

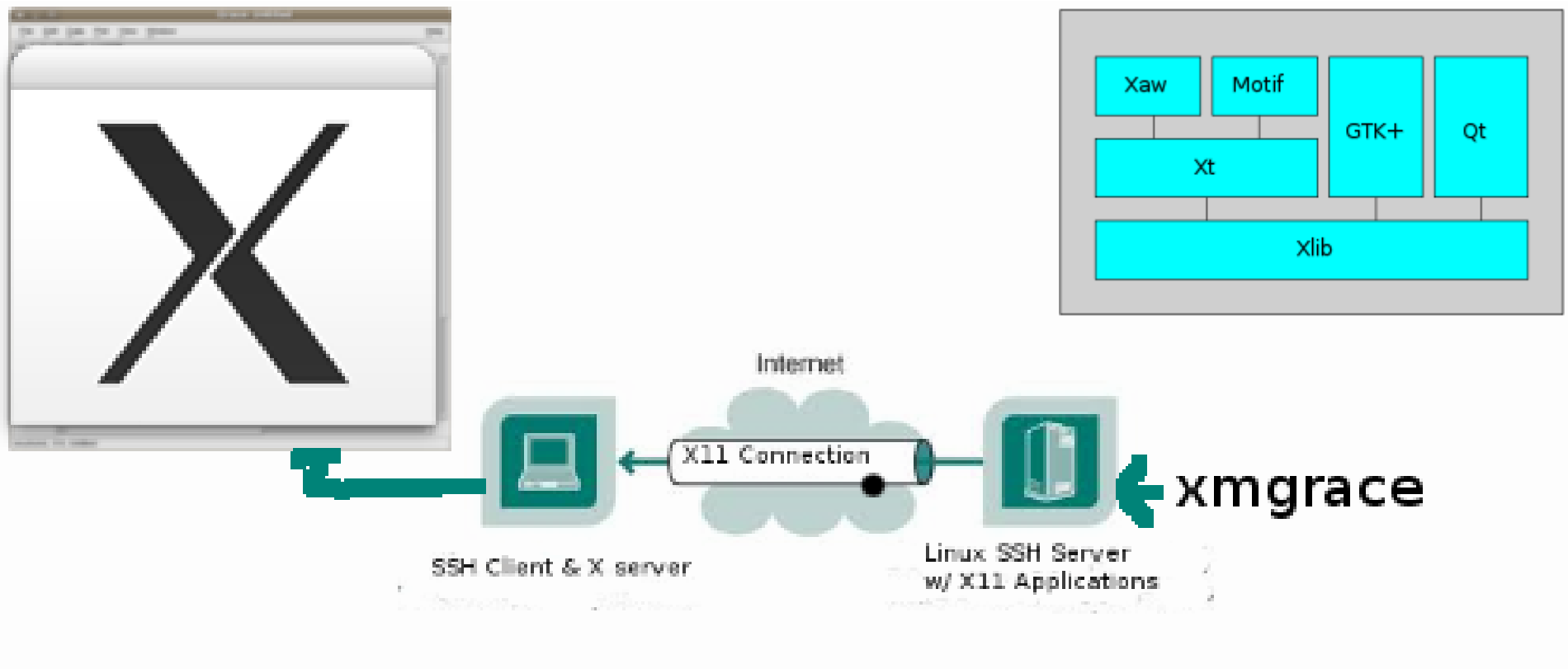


The secure shell (SSH) creates an encrypted “tunnel” between your computer & the Server.



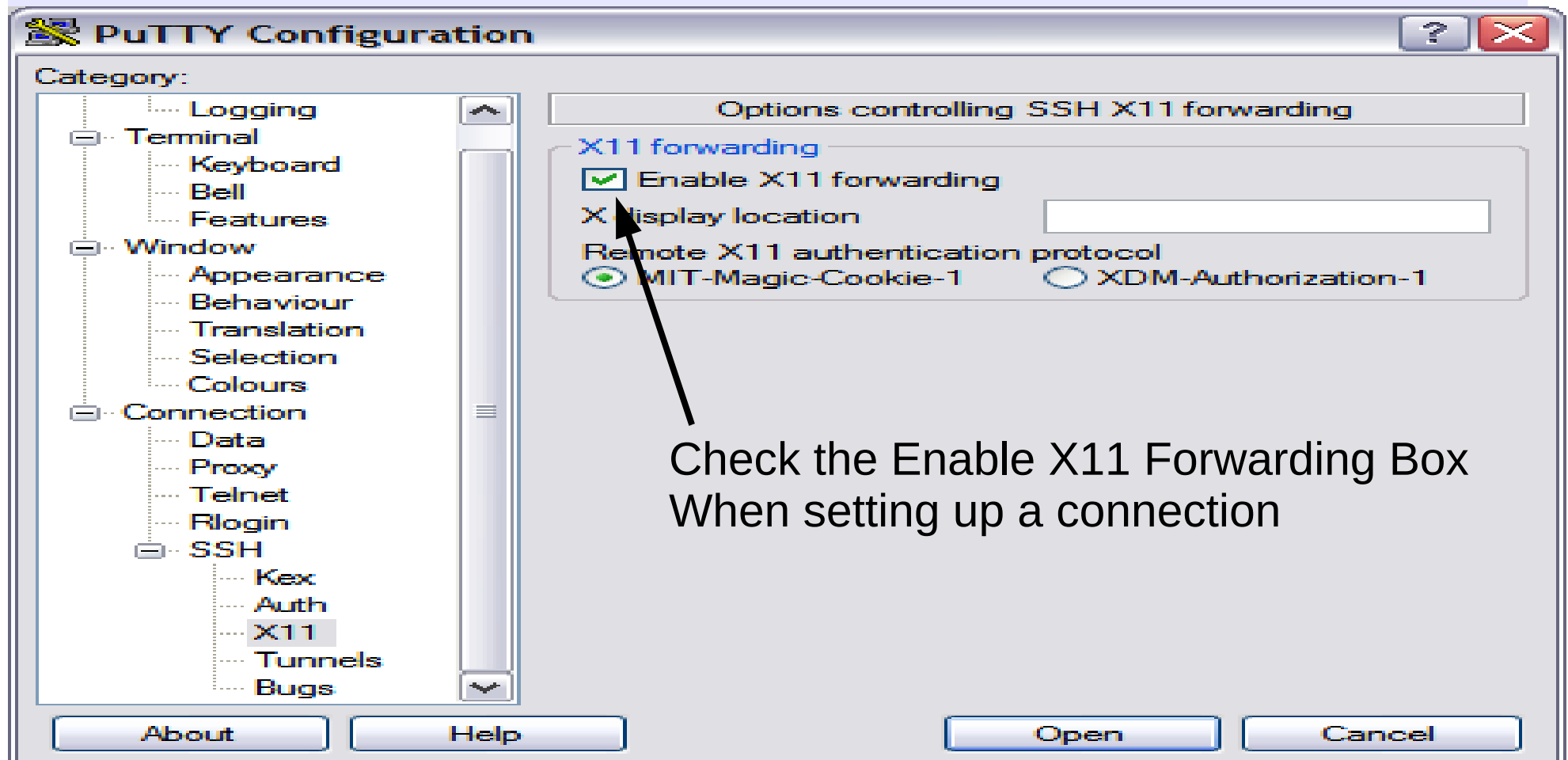
Connecting to NSL Machines Prerequisite 2: An X11 Server

Linux Machines use a library called X11 for graphical interfaces. You need an X Server on your machine to handle this.



SSH clients/X servers available-Windows

puTTY/XMing: Available on isnap.nd.edu website



SSH clients & an X server are installed on Mac OS X & Linux



From a terminal prompt, type

```
ssh -X loginid@machinename
```

File Edit View Terminal Help

```
msuho@msuhond:~$
```

```
msuho@msuhond:~$ ssh -X nsl@regulus.phys.nd.edu
```

```
The authenticity of host 'regulus.phys.nd.edu (129.74.142.141)' can't be established.
```

```
RSA key fingerprint is 10:fa:dd:55:b4:ef:b5:18:46:3b:76:8e:5f:3e:83:f7.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'regulus.phys.nd.edu,129.74.142.141' (RSA) to the list of known hosts.
```

```
nsl@regulus.phys.nd.edu's password:
```

```
Last login: Wed Jul 21 14:15:23 2010 from msuhond.phys.nd.edu
```

```
regulus.phys.nd.edu{nsl}41:
```

You're now on the Linux command line.

**I'm Logged into Linux- Now
What? The Basics.**

A Few Command Line Tips

history - shows a history of all commands typed

!<number>**** - runs command **<number>** from history.

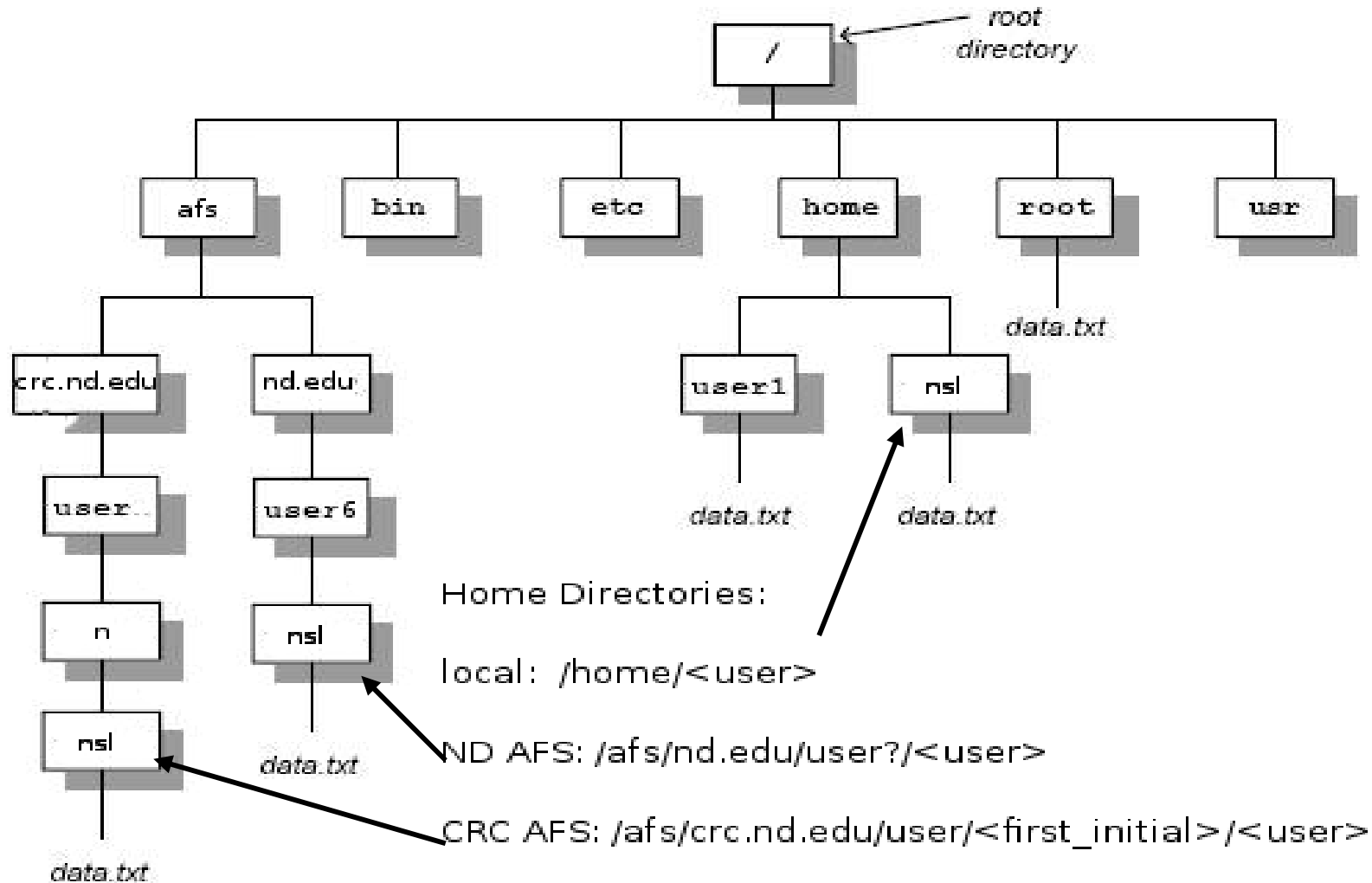
!! - repeats last command

<Tab> autocomplete

Substitution characters: '*' means 'Everything'

exit or **<Control-D>** gets you out.

Directory Structure (it all begins at /) and Your Home Directory



Some Directory Commands

pwd show the directory we're currently in.

cd *<directory path>* changes our current directory.

mkdir [-p] *<directory path>* makes a new directory

rmdir *<directory path>* removes directory (use `rm -r` if nonempty)

A directory can be **absolute** or **relative**. It can also contain special characters, .e.g. `~nsl` is the same as nsl's absolute home directory. `..` refers to your parent directory.

Let's Talk Files

Almost Everything in Linux looks like a file-
data, executable programs, devices, even the
running operating system itself.

Examples:

/var/log/messages	is a data file
/bin/ls	is an executable program
/home/ns1	is a directory
/dev/hda	is a device (hard disk)
/proc/cpuinfo	cpuinfo from running kernel

ls- the king of the file commands

ls [options] [files] – list file content of a directory

- l long listing
- t order output by time instead of lexicographically
- a show hidden files (those starting with a '.')
- r reverse normal sort order (Z-A, Old-New)

File Edit View Terminal Help

```
msuho@msuhond:~/megaraid$ ls -ltra
```

```
total 1000
```

```
-rw-r--r--  1 msuho msuho  1469 2008-03-26 14:38 ut_linux_mgr_5.20.txt
-rw-r--r--  1 msuho msuho 253549 2008-03-26 14:39 ut_linux_mgr_5.20.zip
-rw-r--r--  1 msuho msuho   3512 2008-03-26 14:39 ut_linux_megarc_1.11.txt
-rw-r--r--  1 msuho msuho 301703 2008-03-26 14:40 ut_linux_megarc_1.11.zip
-rw-r--r--  1 msuho msuho   3007 2008-03-26 14:40 3p8_MegaMONr.txt
-rw-r--r--  1 msuho msuho 424814 2008-03-26 14:43 HWR_Monitor_Linux_3.8.zip
drwxr-xr-x  2 msuho msuho   4096 2009-03-27 10:34 .
drwxr-xr-x 139 msuho msuho  12288 2010-07-23 13:43 ..
msuho@msuhond:~/megaraid$ ls
```

Files, owners, & permissions

- A file is owned by some user.
- A file also belongs to one or more groups
- Files, in some combination, may be read, written to, or executed by their owners, groups to which they belong, or someone else.

The **chmod** command allows the owner of a file to change its permissions.

```
msuho@msuhond:~$ ls -l empire.jpg
-rw----- 1 msuho msuho 340255 2009-11-18 09:25 empire.jpg
msuho@msuhond:~$ chmod +r empire.jpg
msuho@msuhond:~$ ls -l empire.jpg
-rw-r--r-- 1 msuho msuho 340255 2009-11-18 09:25 empire.jpg
```

AFS file permissions are different- we'll cover that later.

More File Commands

file <file>

```
msuho@msuhond:/opt/tibs$ file t*
tera:      ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamical
ly linked (uses shared libs), for GNU/Linux 2.6.15, stripped
terad:     ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamical
ly linked (uses shared libs), for GNU/Linux 2.6.15, stripped
teralog.txt: ASCII text
tibs.ini:  regular file, no read permission
```

locate [-b] <file>

```
msuho@msuhond:~$ locate 3p8
/home/msuho/megaraid/3p8_MegaMONr.txt
msuho@msuhond:~$
```

rm [-rf] <files> - remove files

cp [-r] <source><destination> - copy file(s), directory

mv <source> <destination> - moves file or directory

What's a Text File?

A file containing one or more lines of characters. A line feed(LF) ends each line.

```
msuho@msuhond:~/nslkickstart/trunk$ file issue
issue: ASCII English text
msuho@msuhond:~/nslkickstart/trunk$ cat issue

*****
*****
**
**          This system is for the use of AUTHORIZED USERS ONLY.          **
**
**  Individuals using this computer system without authority, or in excess  **
**  of their authority, are subject to having all of their activities on    **
**  this system monitored and recorded by system personnel.                **
**
**  Anyone using this system expressly consents to such monitoring and is   **
**  advised that if such monitoring reveals possible evidence of criminal   **
**  activity, system personnel may provide the evidence of such monitoring  **
**  to law enforcement officials.                                           **
**
**  See the University of Notre Dame Responsible Use Policy for information  **
**  regarding proper use of this machine (http://oit.nd.edu/policies). **
**
*****
*****
```

Displaying Text Files

cat <files>

vs. **more** <files>

```
msuho@msuhond:~$ cat nohuptest.cpp
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    int i=0;
    ofstream ff("test.txt");
    while (i<1000000001)
    {
        if ((i%1000)==0)
        {
            ff << "test    " << i << endl;
        }
        i++;
    }
    ff.close();
    return 1;
}

msuho@msuhond:~$
```

```
msuho@msuhond:~$ more pre10.04pkglist
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                               Version
      Description
+++-----
-----
rc  915resolution                       0.5.3-0ubuntu1
      resolution modification tool for Intel graphic chipset
ii  acpi                                 1.4-2
      displays information on ACPI devices
ii  acpi-support                         0.129
      scripts for handling many ACPI events
ii  acpid                                 1.0.6-9ubuntu8
      Utilities for using ACPI power management
ii  adduser                              3.110ubuntu7
      add and remove users and groups
ii  adobereader-enu                      8.1.1
      Adobe Reader allows you to view navigate and print PDF
ii  aisleriot                            1:2.28.0-0ubuntu1
      Solitaire card games
--More-- (0%)
```

Editing a text File in Linux

Use a *text editor*. Two simple ones are **gedit** & **nano** (shown below). There are many others, including **vi** & **emacs**. Learn to use one.

```
GNU nano 2.2.2          File: ntp.conf

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.

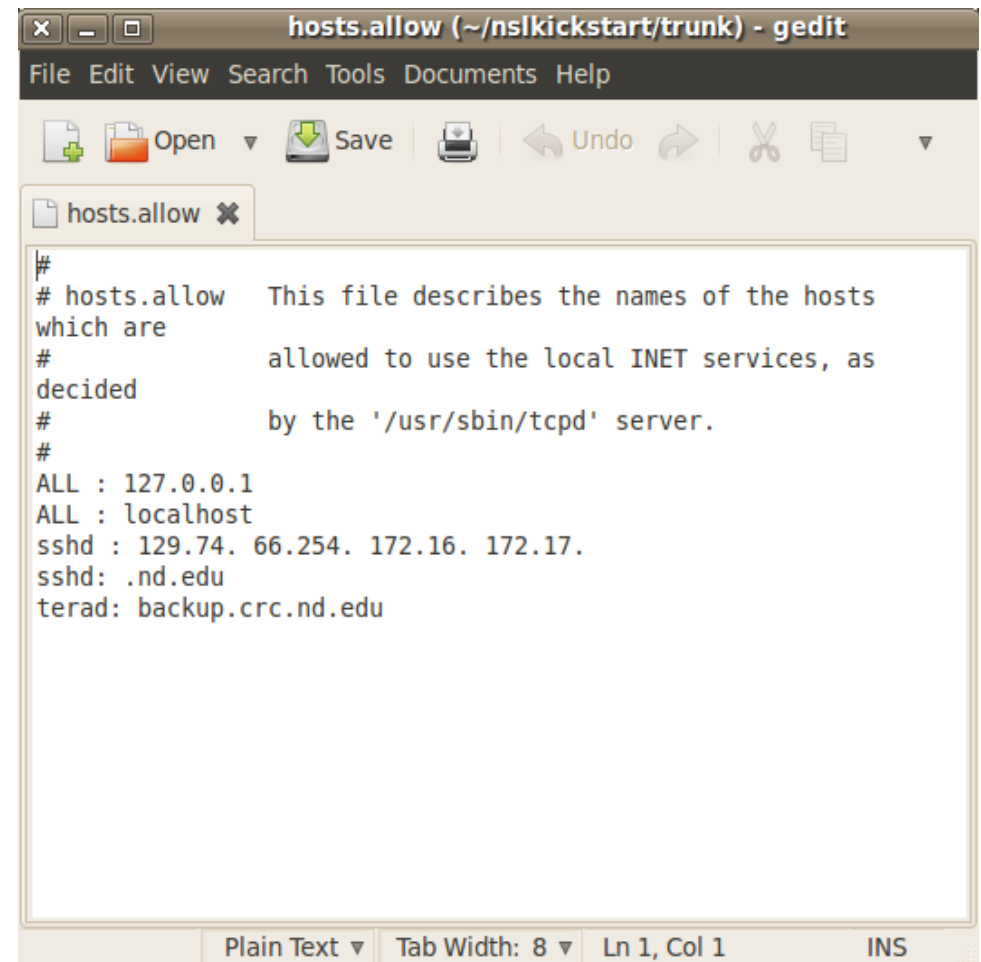
restrict default nomodify notrap noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1

# -- CLIENT NETWORK -----
# Permit systems on this network to synchronize with this
# time service. Do not permit those systems to modify the
# configuration of this service. Also, do not use those
# systems as peers for synchronization.
# restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# --- OUR TIMESERVERS -----

[ Read 67 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```



The screenshot shows the gedit text editor window titled "hosts.allow (~/.ns/kickstart/trunk) - gedit". The window has a menu bar with "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". Below the menu bar is a toolbar with icons for "Open", "Save", "Undo", and "Redo". The main editing area shows the content of the "hosts.allow" file:

```
#
# hosts.allow This file describes the names of the hosts
# which are
#             allowed to use the local INET services, as
#             decided
#             by the '/usr/sbin/tcpd' server.
#
ALL : 127.0.0.1
ALL : localhost
sshd : 129.74. 66.254. 172.16. 172.17.
sshd: .nd.edu
terad: backup.crc.nd.edu
```

At the bottom of the window, there is a status bar showing "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

Displaying Some Other File Types

Use **gv**
(*ghostview*)

Or **evince**
commands to
display PDF,
Postscript, and
EPS file types.

gv: Digital UniverseWEB.qxp

File State Page Portrait 1.000 y792x612 Digital UniverseWEB.qxp D:20070316125751-04'00'

Variable Size
142 x 715

Open
Print All
Print Marked
Save All
Save Marked

<< >>
Reload

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

account for 10% of the digital universe, but will grow 30%-40% faster than mature economies.

- In 2007 the amount of information created will surpass, for the first time, the storage capacity available.

This incredible growth of the digital universe means more than simply the fact that as individuals we will be facing information explosion on an unprecedented scale. It has implications for organizations concerning privacy, security, intellectual property protection, content management, technology adoption, information management, and data center architecture.

The growth and heterogeneous character of the bits in the digital universe mean that organizations worldwide, large and small, whose IT infrastructures transport, store, secure, and replicate these bits, have little choice but to employ ever more sophisticated techniques for information management, security, search, and storage.

HOW DID WE GET THE NUMBERS?

Information about our methodology and underlying assumptions can be found in the section "Methodology and Key Assumptions," but our basic approach was to take IDC forecasts for devices that create or capture digital information – personal computers, digital cameras, servers, sensors, etc. – and estimate the total number of megabytes they capture or produce in a year. We used IDC research and other sources to estimate how much of that data was replicated or copied – as email attachments, archived files, broadcasts, and so on.

Our research follows on previous work conducted at the University of California, Berkeley. Although our methodology varied from that in the Berkeley study – which examined the creation of original information (not including copies) and estimated how much digital information that would represent if all of it were converted to digital format – many of the underlying assumptions were the same."

But our methodology allowed us to size and forecast all the information created and replicated in the digital universe, segment it by region, and put it in context with the available storage capacity. We believe ours is the first-ever study to size and forecast the rate of expansion of the entire digital universe.

WHAT ARE BITS AND BYTES?

A "bit" is the smallest unit of information that can be stored in a computer, and consists of either a 1 or 0 (or on/off state). All computer calculations are in bits.

A "byte" is a collection of 8 bits. Bytes are convenient because, when converted to computer code, they can represent 256 characters, such as numbers or letters. So a byte is 8 times larger than a bit.

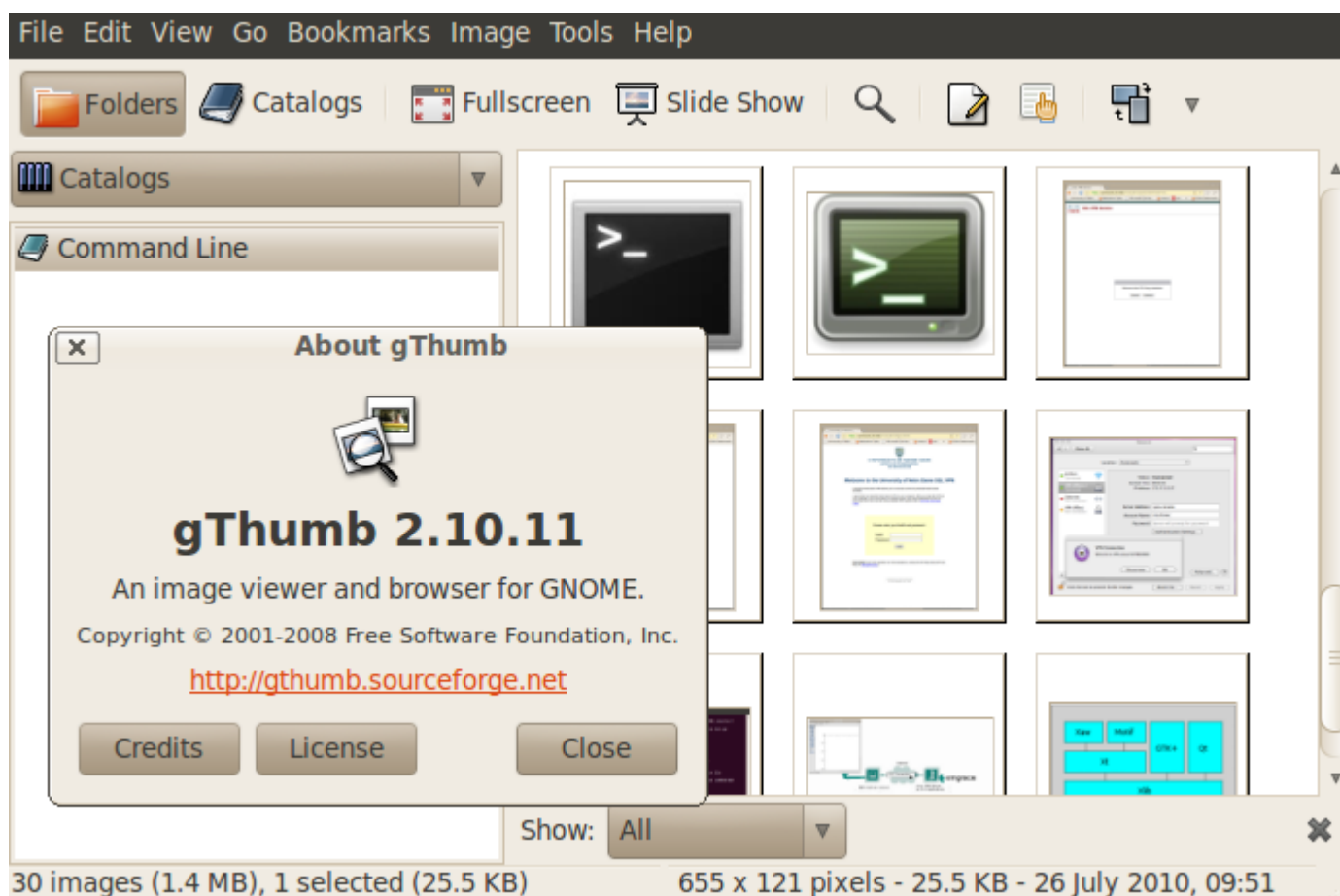
Common aggregations for bytes come in multiples of 1,000, such as kilobyte, megabyte, gigabyte, and so on. The progression is as follows:

Bit (b)	1 or 0
Byte (B)	8 bits
Kilobyte (KB)	1,000 bytes
Megabyte (MB)	1,000 KB
Gigabyte (GB)	1,000 MB
Terabyte (TB)	1,000, GB
Petabyte (PB)	1,000 TB
Exabyte (EB)	1,000 PB
Zettabyte (ZB)	1,000 EB

This seems simple enough, except sometimes multiples of bytes are considered as powers of 2, since the original machine language only has two states, 1 or 0. A kilobyte would then be 2^{10} bytes, or 1,024 bytes. A megabyte would be 2^{20} bytes, or 1,024 kilobytes, and so on.

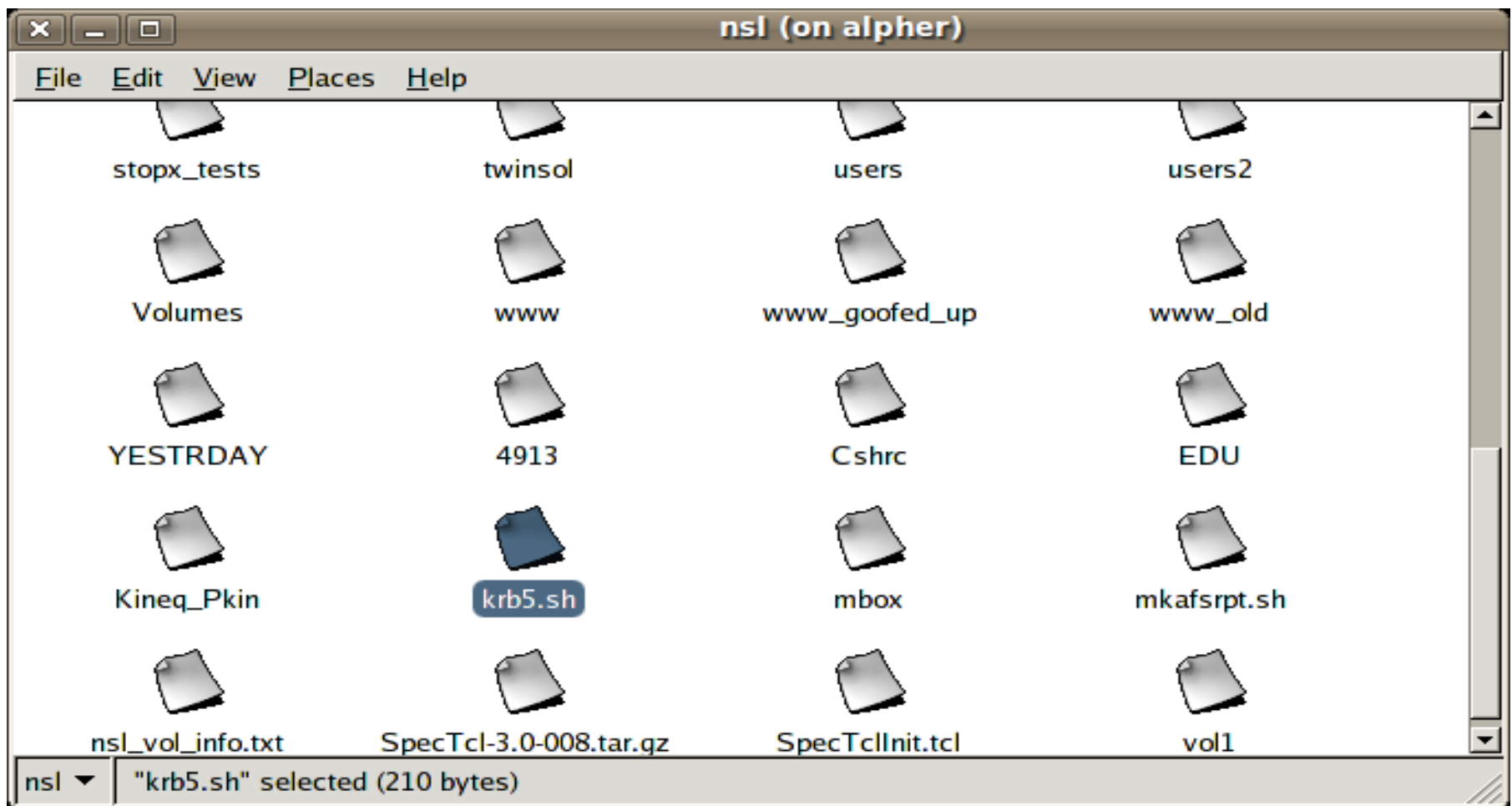
For the sake of simplicity, in all calculations for this research we used the decimal system we mentioned first. This is consistent with the representation used in the Berkeley study.

Use **gthumb** for displaying gifs, jpgs, pngs & other common image files



nautilus – a GUI File Browser

nautilus, typed from the command line, starts a GUI file browser like Windows Explorer or MAC Finder. Right click on a file & try “open with”

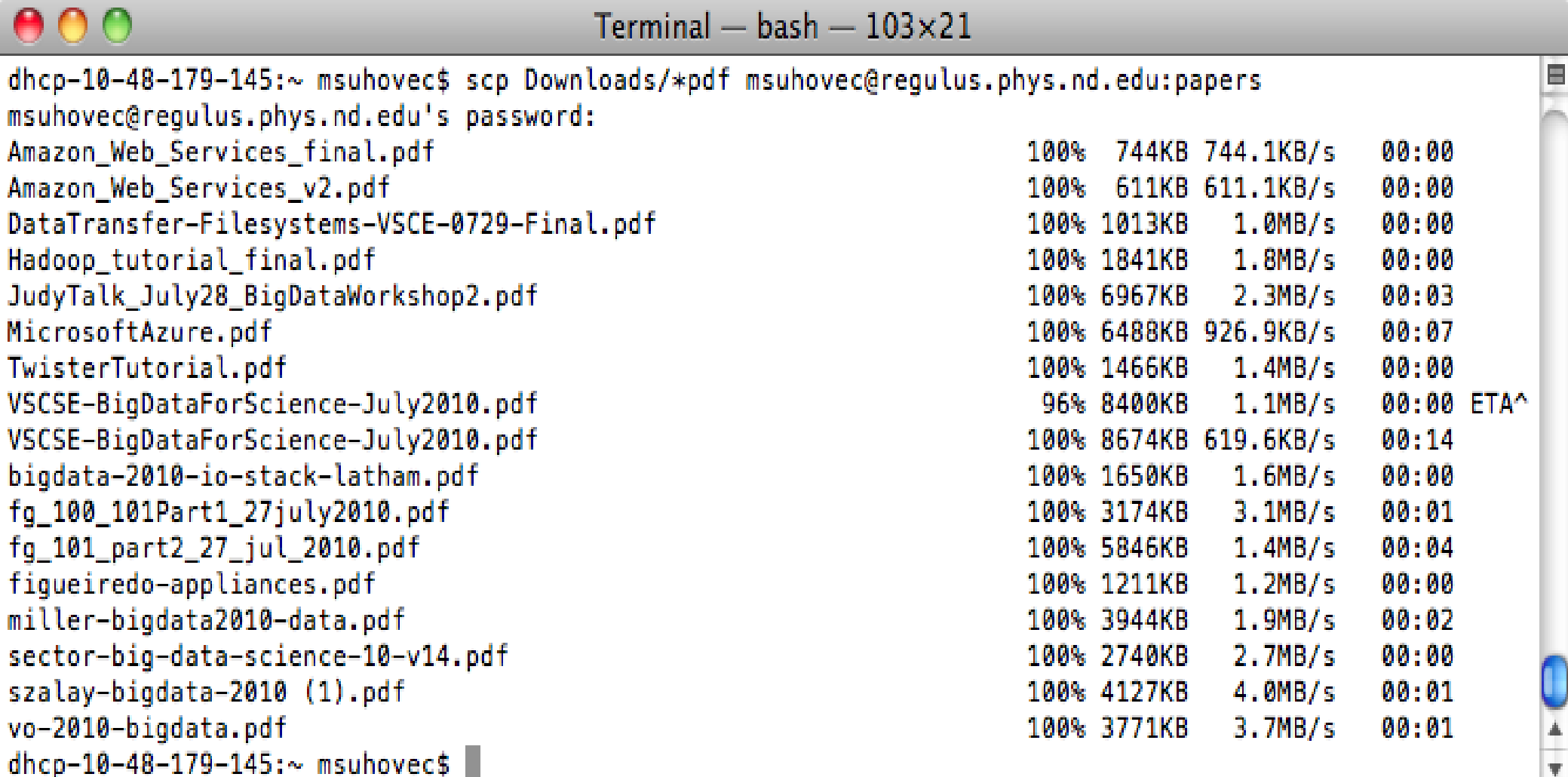


Uploading & Downloading Files, Dealing with File Archives

Copying Files with scp

Secure Copy (scp) uses SSH to copy files between machines

```
scp [-r] <from> <to>
```



```
Terminal — bash — 103x21
dhcp-10-48-179-145:~ msuhovec$ scp Downloads/*.pdf msuhovec@regulus.phys.nd.edu:papers
msuhovec@regulus.phys.nd.edu's password:
Amazon_Web_Services_final.pdf          100% 744KB 744.1KB/s  00:00
Amazon_Web_Services_v2.pdf            100% 611KB 611.1KB/s  00:00
DataTransfer-Filesystems-VSCE-0729-Final.pdf 100% 1013KB 1.0MB/s  00:00
Hadoop_tutorial_final.pdf             100% 1841KB 1.8MB/s  00:00
JudyTalk_July28_BigDataWorkshop2.pdf  100% 6967KB 2.3MB/s  00:03
MicrosoftAzure.pdf                   100% 6488KB 926.9KB/s 00:07
TwisterTutorial.pdf                   100% 1466KB 1.4MB/s  00:00
VSCSE-BigDataForScience-July2010.pdf   96% 8400KB 1.1MB/s  00:00 ETA^
VSCSE-BigDataForScience-July2010.pdf  100% 8674KB 619.6KB/s 00:14
bigdata-2010-io-stack-latham.pdf       100% 1650KB 1.6MB/s  00:00
fg_100_101Part1_27july2010.pdf         100% 3174KB 3.1MB/s  00:01
fg_101_part2_27_jul_2010.pdf           100% 5846KB 1.4MB/s  00:04
figueiredo-appliances.pdf              100% 1211KB 1.2MB/s  00:00
miller-bigdata2010-data.pdf             100% 3944KB 1.9MB/s  00:02
sector-big-data-science-10-v14.pdf     100% 2740KB 2.7MB/s  00:00
szalay-bigdata-2010 (1).pdf             100% 4127KB 4.0MB/s  00:01
vo-2010-bigdata.pdf                     100% 3771KB 3.7MB/s  00:01
dhcp-10-48-179-145:~ msuhovec$
```

wget :Download Files from the Internet

wget <file URI>

```
File Edit View Terminal Help

regulus.phys.nd.edu{ns1}60: wget ftp://root.cern.ch/root/root_v5.26.00c.source.t
ar.gz
--2010-07-15 14:12:35--  ftp://root.cern.ch/root/root_v5.26.00c.source.tar.gz
      => `root_v5.26.00c.source.tar.gz'
Resolving root.cern.ch... 128.142.172.95
Connecting to root.cern.ch|128.142.172.95|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.      ==> PWD ... done.
==> TYPE I ... done.   ==> CWD /root ... done.
==> SIZE root_v5.26.00c.source.tar.gz ... 28922182
==> PASV ... done.     ==> RETR root_v5.26.00c.source.tar.gz ... done.
Length: 28922182 (28M)

100%[=====>] 28,922,182    214K/s   in 22s

2010-07-15 14:13:00 (1.24 MB/s) - `root_v5.26.00c.source.tar.gz' saved [28922182
]

regulus.phys.nd.edu{ns1}61: ls
root_v5.26.00c.source.tar.gz
regulus.phys.nd.edu{ns1}62:
regulus.phys.nd.edu{ns1}62:
regulus.phys.nd.edu{ns1}62:
```

Dealing with Linux File Archives

Files ending **.tar**, **.tar.gz**, or **tar.bz2** are archives of multiple files, bundled up for easy transport.

Use the **tar** command to manage these:

tar t[vzj]f <tar file> shows contents without unbundling.

tar x[vzj]f <tar file> extracts files/directories from archive

Options:

v – verbose

z - unzip zip files

j - unzip bzip2 files

tar example 1

```
msuho@msuhond:~$ tar tvf Kineq_Pkin.tar
-rw-r--r-- nsl/campus 817664 2010-03-19 11:01 Kineq_Pkin/Makefile
-rw-r--r-- nsl/campus 122 1997-08-06 11:42 Kineq_Pkin/double_precision.f90
-rw-r--r-- nsl/campus 3525 1997-08-06 11:44 Kineq_Pkin/get_mass.f90
-rw-r--r-- jkaiser/other 371250 2001-07-17 12:59 Kineq_Pkin/mass.copy
-rw-r--r-- nsl/campus 371250 1997-02-04 09:45 Kineq_Pkin/mass.copy.orig
-rw-r--r-- nsl/campus 1450 1997-08-06 11:43 Kineq_Pkin/nuc_constants.f90
-rw-r--r-- nsl/campus 23219 1997-08-06 11:43 Kineq_Pkin/parser.f90
-rw-r--r-- nsl/campus 26134 1997-08-06 11:44 Kineq_Pkin/pkin.f90
-rw-r--r-- jkaiser/campus 0 2001-05-25 10:58 Kineq_Pkin/pkin.log
-rw-r--r-- nsl/campus 11092 1997-08-06 11:43 Kineq_Pkin/relkinmat.f90
-rw-r--r-- nsl/campus 931 1997-08-06 11:43 Kineq_Pkin/z_elements.f90
msuho@msuhond:~$
```

tar example 2

```
regulus.phys.nd.edu{ns1}57: tar xzvf dawn_3_88a.taz | more
dawn_3_88a/
dawn_3_88a/socket.aix
dawn_3_88a/README.txt
dawn_3_88a/FRMark.cc
dawn_3_88a/main_selector2.cc
dawn_3_88a/parameter.cc
dawn_3_88a/g4test_inet.cc
dawn_3_88a/hidA.cc
dawn_3_88a/parameter.h
dawn_3_88a/FRVector2.h
dawn_3_88a/FRPolycone.h
dawn_3_88a/FRSocketMacro.h
dawn_3_88a/configure_min
dawn_3_88a/config_cygnus.sh
dawn_3_88a/hid2.cc
dawn_3_88a/attribute.h
dawn_3_88a/FRParallelepipedForBrep.cc
dawn_3_88a/command2.cc
dawn_3_88a/command.h
dawn_3_88a/object3.cc
dawn_3_88a/list.h.1
dawn_3_88a/vector.h
dawn_3_88a/linesegment.cc
```

Unzipping files

If file ends with a `.zip`, use **unzip**. If file ends with a `.gz`, use **gunzip**

```
msuho@msuhond:~/example$ ls
divbasedtable-2009-01-13.zip
msuho@msuhond:~/example$ unzip divbasedtable-2009-01-13.zip
Archive:  divbasedtable-2009-01-13.zip
  inflating: DIVBasedTable.inc.php
  inflating: index.php
msuho@msuhond:~/example$
```

```
msuho@msuhond:~/Downloads$ ls include*
include.tar.gz
msuho@msuhond:~/Downloads$ gunzip include.tar.gz
msuho@msuhond:~/Downloads$ ls include*
include.tar
msuho@msuhond:~/Downloads$
```

Living with AFS

What is AFS?

The Andrew File System (AFS) is a distributed file System developed at Carnegie-Mellon University in the 1980s. It's a global file system with many **cells** :

```
msuho@msuhond:/afs$ ls /afs
andrew.cmu.edu  crc.nd.edu      ncsa.uiuc.edu  pitt.edu        umr.edu
anl.gov        fnal.gov        nd.edu         psc.edu
athena.mit.edu grandcentral.org nersc.gov      slac.stanford.edu
cern.ch        msu.edu         openafs.org    umich.edu
msuho@msuhond:/afs$
```

Notre Dame currently has two AFS cells:
nd.edu & crc.nd.edu

Your AFS Home Directory

Anyone with a valid ND netId has an AFS home directory in the nd.edu AFS cell. The nd.edu common AFS password file is ***/afs/nd.edu/common/etc/passwd***

```
msuho@msuhond:/afs$ grep msuhovec /afs/nd.edu/common/etc/passwd
msuhovec:X:124872:40:Mark Suhovecky:/afs/nd.edu/user19/msuhovec:/bin/csh
msuho@msuhond:/afs$
```

Anyone with a valid Center for Research Computing (CRC) account has an AFS home directory in the crc.nd.edu AFS cell. The crc.nd.edu common AFS password file is ***/afs/crc.nd.edund.edu/common/etc/passwd***

```
msuho@msuhond:/afs$ grep msuhovec /afs/crc.nd.edu/common/etc/passwd
msuhovec:X:124872:1313:Mark Suhovecky:/afs/crc.nd.edu/user/m/msuhovec:/bin/tcsh
msuho@msuhond:/afs$
```

NSL AFS Group Volumes

5 research programs in the lab each have volumes for keeping experimental data. Don't keep experimental data in your private directory!

nd.edu - /afs/nd.edu/nsi/Volumes

crc.nd.edu - /afs/crc.nd.edu/group/nsi

Group Names: **ast** (Wiescher) , **ams** (Collon),
gam(Garg), **nuc** (Aprahamian), **react** (Tang)

AFS Authentication & Access Lists

To use AFS, you must be authenticated (AFS knows who you are) and authorized (You've got an authorization **token** from AFS).

If you logged into an AFS directory, this was done for you at login time.

If not, or if you're switching between different accounts, you do these two steps with the **kinit** & **aklog** commands.

The **kinit** command talks to a University Kerberos server to authenticate with your netID.

aklog talks to AFS to get a token.

AFS Authentication Commands

kinit <netID>@<domain> Authenticates user (Domain is ND.EDU or CRC.ND.EDU)

klist Shows your current authentication

kdestroy clears your current authentication

aklog <domain> Gets an authenticated user an AFS token (Domain is ND.EDU or CRC.ND.EDU).

tokens Shows your current AFS tokens.

unlog Clears your current AFS tokens.

kinit/aklog Example

```
msuho@msuhond:/afs/nd.edu/nsl/Volumes$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_124872)
msuho@msuhond:/afs/nd.edu/nsl/Volumes$ tokens

Tokens held by the Cache Manager:

--End of list--
msuho@msuhond:/afs/nd.edu/nsl/Volumes$ kinit msuhovec@ND.EDU
Password for msuhovec@ND.EDU:
msuho@msuhond:/afs/nd.edu/nsl/Volumes$ aklog ND.EDU
msuho@msuhond:/afs/nd.edu/nsl/Volumes$ klist
Ticket cache: FILE:/tmp/krb5cc_124872
Default principal: msuhovec@ND.EDU

Valid starting    Expires          Service principal
08/02/10 13:07:12 08/03/10 13:07:12 krbtgt/ND.EDU@ND.EDU
08/02/10 13:07:45 08/03/10 13:07:12 afs@ND.EDU
msuho@msuhond:/afs/nd.edu/nsl/Volumes$ tokens

Tokens held by the Cache Manager:

User's (AFS ID 124872) tokens for afs@nd.edu [Expires Aug  3 13:07]
--End of list--
msuho@msuhond:/afs/nd.edu/nsl/Volumes$
```

AFS Space & ACLs commands

fs listquota <directory> Shows space usage information for AFS volume where <directory> resides.

fs listacl <directory> Shows Access Control List (AFS permissions) for <dir>.

AFS Space/ACL Example

```
regulus.phys.nd.edu{ns1}41: cccdr
regulus.phys.nd.edu{ns1}42: fs listquota /afs/nd.edu/ns1/Volumes/ns1.ams.4
Volume Name          Quota      Used %Used  Partition
ns1.ams.4            20000000  5800345  29%      81%
regulus.phys.nd.edu{ns1}43: fs listacl /afs/nd.edu/ns1/Volumes/ns1.ams.4
Access list for /afs/nd.edu/ns1/Volumes/ns1.ams.4 is
Normal rights:
  ns1:ams rlidwka
  system:administrators rlidwka
  jkaiser rlidwka
  llamm rlidwka
  ns1 rlidwka
  ams rlidwka
  kpost rlidwk
  mbowers2 rlidwk
regulus.phys.nd.edu{ns1}44: █
```

r – read, **l** -list, **i**- insert, **d**- delete, **w**-write, **k**- lock, **a**- administer
rl = read, **rlidwk** = read/write

Using webfile.nd.edu with AFS



Manage AFS Access Rights at
nsl/Volumes/nsl.ams.6.pcollon

Change Existing ACLs at nsl/Volumes/nsl.ams.6.pcollon

User/Group	Access Rights	Change To
jkaiser	Read	(please select one) ▼
nsl	All	(please select one) ▼
pcollon	All	(please select one) ▼
drobert4	All	(please select one) ▼
kpost	Write	(please select one) ▼

Using Modules

Every User's Linux Environment can be Different

```
[nsl@arcturus ~]$ env
```

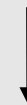
```
USER=nsl
```

```
LOGNAME=nsl
```

```
HOME=/afs/nd.edu/user6/nsl
```

```
PATH=/usr/nuclear/grace/bin:/opt/und/intel/x86_64/fc/10.0.023/bin:/opt/und/intel/x86_64/cc/10.0.023/bin:/opt/und/intel/x86_64/idb/10.0.023/bin:/opt/und/schrodinger/2007:/opt/und/schrodinger/2007/utilities:/opt/und/local/bin:/opt/und/mathematica/6.0/Executables:/opt/und/maple/12/bin:/opt/und/pgi/6.1-2/linux86-64/6.1/bin:/afs/nd.edu/i386_linux24/opt/und/matlab/7.7/bin:/opt/und/modules/3.1.6/bin:/usr/lib64/qt-3.3/bin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/usr/X11R6/lib:./opt/tibs:/usr/nuclear/CERN/2002/bin:/usr/nuclear/hhirf:/usr/nuclear/rw01/bin:/usr/nuclear/bin:/usr/nuclear/lib:/usr/nuclear/transport:/usr/nuclear/CERN/root/bin:./usr/nuclear/rp/bin:.
```

Yikes!



Modules keep the Linux Environment Manageable

- Include only what config info you need.
- Configuration info need only reside in one place.

Three commands:

module list - See what's currently loaded for you

module avail – See all available modules

module load/unload <module> - load/unload module

module list Example

```
[nsl@arcturus ~]$ module list
Currently Loaded Modulefiles:
  1) modules
  2) matlab/7.7
  3) pgi/6.1-2
  4) maple/12
  5) mathematica/6.0
  6) opt_local/1.0
  7) schrodinger/2007
  8) initialize/standard
  9) intel/10.0
 10) grace/5.1.21
[nsl@arcturus ~]$
```

module avail Example

```
[nsl@arcturus ~]$ module avail
```

```
----- /usr/nuclear/modulefiles -----  
daphne      geant/4.9.0  mcnp/mcnp5   root/5.22  
dawn/3.88a   geant/4.9.3  paw/2003     root/5.26  
gawk/3.1.6   grace/5.1.21 root/5.16    spectcl
```

module load Example

```
[nsl@arcturus ~]$ module load root
```

```
[nsl@arcturus ~]$ module list
```

```
Currently Loaded Modulefiles:
```

- | | |
|--------------------|------------------------|
| 1) modules | 7) schrodinger/2007 |
| 2) matlab/7.7 | 8) initialize/standard |
| 3) pgi/6.1-2 | 9) intel/10.0 |
| 4) maple/12 | 10) grace/5.1.21 |
| 5) mathematica/6.0 | 11) root/5.26 |
| 6) opt local/1.0 | |

Other Helpful Linux Things

Redirecting Command Output

The `>` operator saves the command output to a file.

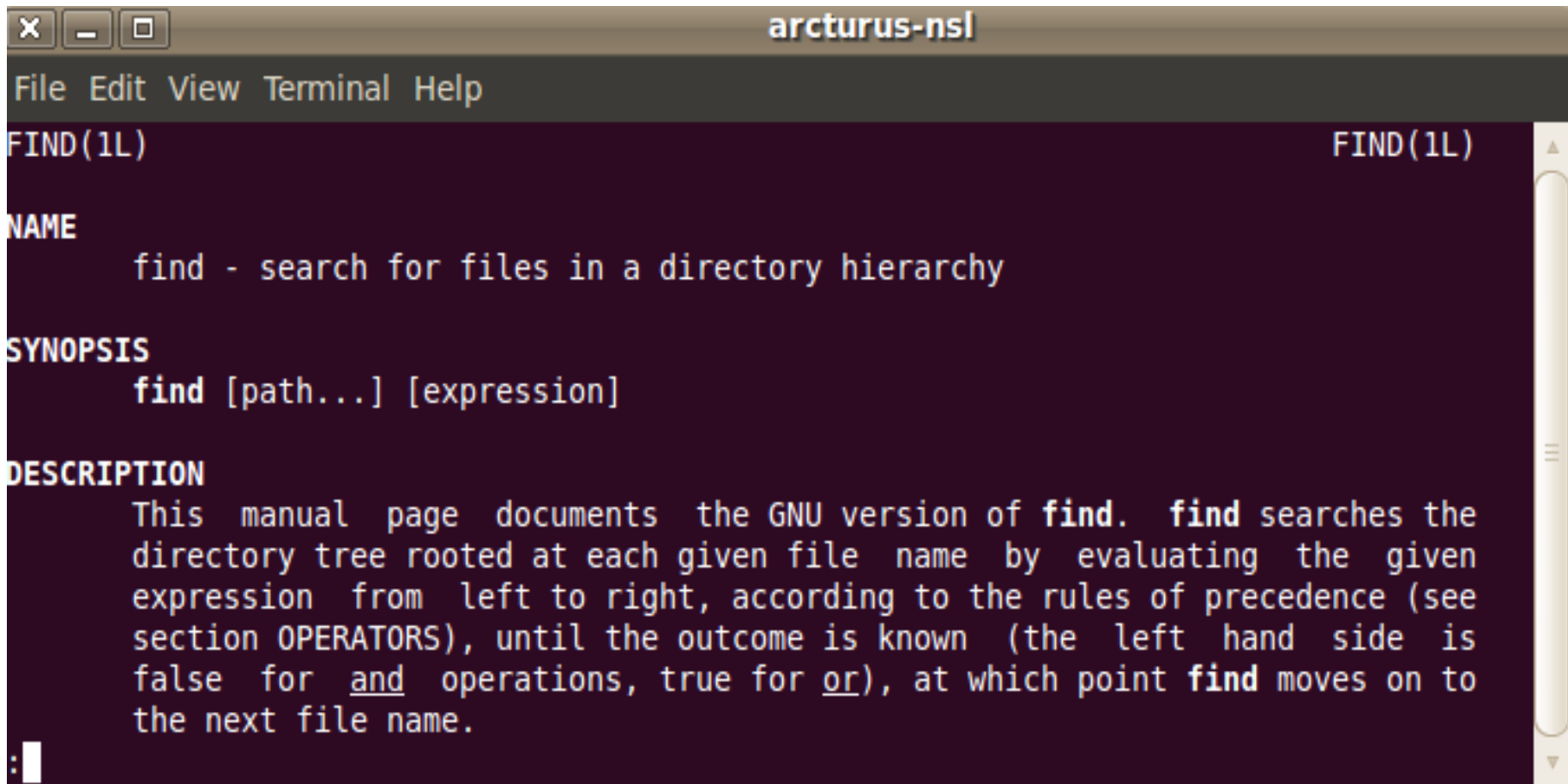
```
msuho@msuhond:~$ ls -l *zip > ls_long.txt
msuho@msuhond:~$ cat ls_long.txt
-rw-r--r-- 1 msuho msuho 31922768 2009-12-04 11:29 lastiki_trunk.zip
```

The `|` operator ('pipe') sends one command's output another command's input. This is incredibly useful- you create new functionality by stringing existing commands together.

```
msuho@msuhond:~$ ls -l *zip
-rw-r--r-- 1 msuho msuho 31922768 2009-12-04 11:29 lastiki_trunk.zip
msuho@msuhond:~$ ls -l *zip | cut -c25-33
31922768
msuho@msuhond:~$
```

How do I use that Command, Again?

The **man** command will show you a manual page for almost any Linux command. The **info** command will, too.



```
arcturus-ns1
File Edit View Terminal Help
FIND(1L) FIND(1L)
NAME
  find - search for files in a directory hierarchy
SYNOPSIS
  find [path...] [expression]
DESCRIPTION
  This manual page documents the GNU version of find. find searches the
  directory tree rooted at each given file name by evaluating the given
  expression from left to right, according to the rules of precedence (see
  section OPERATORS), until the outcome is known (the left hand side is
  false for and operations, true for or), at which point find moves on to
  the next file name.
```

Searching for stuff with **find**

The **find** command will search for (& perform actions on) files matching criteria you specify (file name, age, size & almost anything else)

```
[nsl@arcturus geant4]$ find /usr/nuclear/geant -atime -1
/usr/nuclear/geant/4.9.3/src/geant4
/usr/nuclear/geant/4.9.3/src/geant4/examples/novice/N02
/usr/nuclear/geant/4.9.3/src/geant4/Configure
```

```
[nsl@arcturus geant4]$ find /usr/nuclear/geant -type d -regex '.*Radioactive.*'
/usr/nuclear/geant/4.9.0/data/RadioactiveDecay3.2
/usr/nuclear/geant/4.9.3/data/RadioactiveDecay3.2
```

There are dozens of other options- use the **man** page!

Searching for stuff with **grep**

grep [-r] [-v] [-n] <pattern> <files> searches for <pattern> within <files>. Many other options.

Example: search system logs for successful mcouder logins

```
[root@arcturus]$ grep -r -n mcouder /var/log | grep password
/var/log/secure.1:747:Jul 28 21:02:19 arcturus sshd[29304]: Accepted password fo
r mcouder from ::ffff:129.74.143.155 port 48861 ssh2
/var/log/secure.4:38:Jul  5 10:16:26 arcturus sshd[4655]: Accepted password for
mcouder from ::ffff:129.74.143.155 port 45153 ssh2
/var/log/secure:197:Aug  3 15:49:15 arcturus sshd[17953]: Accepted password for
mcouder from ::ffff:129.74.143.155 port 50352 ssh2
/var/log/secure:504:Aug  6 09:23:57 arcturus sshd[14853]: Accepted password for
mcouder from ::ffff:129.74.143.155 port 34757 ssh2
/var/log/secure.3:226:Jul 12 13:13:12 arcturus sshd[5150]: Accepted password for
mcouder from ::ffff:129.74.143.155 port 45690 ssh2
[root@arcturus]$
```

Who's online?

w – show who's on line

```
[nsl@alpher ~]$ w
 16:27:52 up 37 days, 6:04,  4 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
qli3     pts/0    129.74.141.181  10:05   1:26m  0.34s  0.20s -csh
qli3     pts/1    129.74.141.181  10:46   5:38m  0.10s  0.10s -csh
rnb      pts/3    129.74.141.33   14:58   20:25  0.11s  0.00s ptolemy
nsl      pts/4    msuhond.phys.nd. 16:27   0.00s  0.10s  0.00s w
[nsl@alpher ~]$
```

Running Looooong Programs w/ **nohup**

The **nohup** command lets a command keep running after you've exited (**hung up**)

The **pagsh** command sets up a separate authentication group for AFS tokens.

Used together, one can run long term jobs while offline.

```
[msuhovec@alpher]$ pagsh
[msuhovec@alpher]$ kinit msuhovec@ND.EDU
Password for msuhovec@ND.EDU:
[msuhovec@alpher]$ aklog ND.EDU
[msuhovec@alpher]$ nohup sar 5 60 > sar_nohup.out &
[1] 17031
[msuhovec@alpher]$ exit
```

processes: monitoring, killing

The **ps** command shows statistics about running processes.

The **kill** command will terminate a process (you can only kill one of your own.)

```
[msuhovec@alpher]$ ps -p 17031
  PID TTY          TIME CMD
17031 ?                00:00:00 sar
[msuhovec@alpher]$ ps -lp 17031
F S    UID      PID  PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
0 S 124872 17031    1   0   76   0 - 1297 pipe_w ?        00:00:00 sar
[msuhovec@alpher]$ kill -9 17031
[msuhovec@alpher]$ ps -lp 17031
F S    UID      PID  PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
```

More `ps` examples

`ps -af` will show all running user processes

`ps -fC <cmd>` shows all running `<cmd>`s

`ps -ef` will show everything!

```
[msuhovec@alpher]$ ps -af
UID          PID    PPID  C  STIME TTY          TIME CMD
qli3         16649      1   0  10:10 pts/1        00:00:00 /usr/nuclear/pgplot//pgxwin_serv
msuhovec    19054   17041   0  11:00 pts/0        00:00:00 ps -af
```

```
[msuhovec@alpher]$ ps -fC sshd
UID          PID    PPID  C  STIME TTY          TIME CMD
root          3607      1   0  Jun28 ?           00:00:15 /usr/sbin/sshd
root         16609    3607   0  10:09 ?           00:00:00 sshd: qli3 [priv]
qli3         16615   16609   0  10:10 ?           00:00:05 sshd: qli3@pts/1
root         17036    3607   0  10:51 ?           00:00:00 sshd: msuhovec [priv]
msuhovec     17039   17036   0  10:51 ?           00:00:00 sshd: msuhovec@pts/0
```

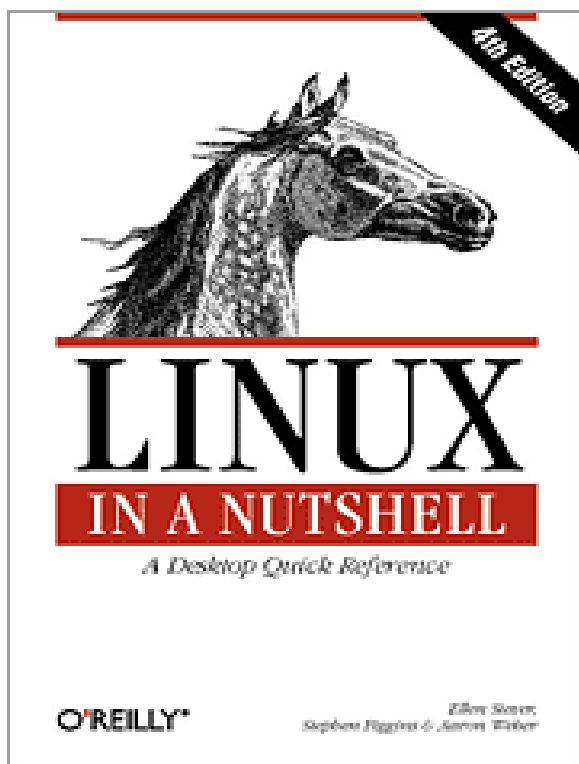
date, cal, and uptime

Give date/time, calendar, and system uptime

```
[msuhovec@alpher]$ date
Fri Aug  6 13:46:33 EDT 2010
[msuhovec@alpher]$ cal
      August 2010
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

[msuhovec@alpher]$ uptime
13:46:46 up 39 days,  3:23,  2 users,  load average: 0.03, 0.02, 0.00
```

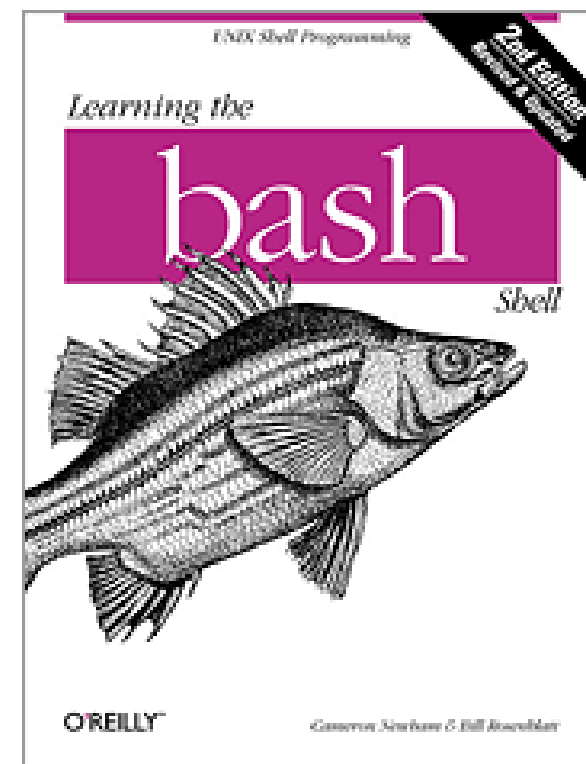
We've Barely Scratched the Surface



Learn More.

There's the Internet-

Books...



ND CRC Training:

http://crcmedia.hpcc.nd.edu/wiki/index.php/Introduction_to_Unix/Linux

Questions?

