

## What is This Thing Called UNIX?

---

Managing your Voyager System Series

# Copyright Statement

All of the information and material inclusive of text, images, logos, product names is either the property of, or used with permission by Ex Libris Ltd. The information may not be distributed, modified, displayed, reproduced – in whole or in part – without the prior written permission of Ex Libris Ltd.

## **TRADEMARKS**

Ex Libris, the Ex Libris logo, Aleph, SFX, SFXIT, MetaLib, DigiTool, Verde, Primo, Voyager, MetaSearch, MetaIndex and other Ex Libris products and services referenced herein are trademarks of Ex Libris, and may be registered in certain jurisdictions. All other product names, company names, marks and logos referenced may be trademarks of their respective owners.

## **DISCLAIMER**

The information contained in this document is compiled from various sources and provided on an "AS IS" basis for general information purposes only without any representations, conditions or warranties whether express or implied, including any implied warranties of satisfactory quality, completeness, accuracy or fitness for a particular purpose.

Ex Libris, its subsidiaries and related corporations ("Ex Libris Group") disclaim any and all liability for all use of this information, including losses, damages, claims or expenses any person may incur as a result of the use of this information, even if advised of the possibility of such loss or damage.

# Agenda

---

- **What is an operating system?**
- Logging in
- The online manual
- Directory basics
- File system hierarchy
- Users and groups
- File system commands
- The shell
- Process management
- Recap & Resources

# What is an Operating System?

---

- A set of programs
- Interface between users and hardware
- Interface between applications and hardware
- Take care of storage management
- Take care of I/O device management

# UNIX History

---

- Forty years after its creation, UNIX is still regarded as one of the most powerful, versatile and flexible operating systems in the computer world.
  - Created at Bell Labs in 1970
  - Written in the C programming language which was developed at the same time
  - Supports large numbers of simultaneous users
  - Runs with few alterations on many hardware platforms
  - Simple, elegant, and easy to use (at least compared to its predecessors)

# Linux Background

- In more recent UNIX history, first developed in 2001, a UNIX variant called Linux has become very popular for a number of reasons.
  - It runs on standard desktop hardware, both PC and Mac
  - It is Free, Open Source Software, meaning both that it is available for no charge and that anyone is allowed to modify it without restriction
  - It has a large support base, including hundreds of free, professional-quality applications
  - It offers an alternative to the Microsoft desktop monopoly

# Terminology

---

- File – bits on disk or tape
- Program – executable, binary
- Script – ASCII file (editable) talks to shell
- Shell – command interpreter (UI); talks to kernel
- Kernel – core or key components of the O.S.; talks to hardware

# Agenda

---

- What is an operating system?
- **Logging in**
- The online manual
- Directory basics
- File system hierarchy
- Users and groups
- File system commands
- The shell
- Process management
- Recap & Resources



# Logging In

- SSH – “Secure Shell” Telnet
- Identifying yourself with a username and password
- The Command Line Interface (CLI)
- Don’t get the Linux and the DOS CLIs confused
- The CLI prompt is configurable:

```
[VGER] voyager@dc02vg0141na : logs/ =>
```



voyager.coalliance.org - PuTTY

login as: voyager

voyager@voyager.coalliance.org's password:

Last login: Wed Nov 2 09:51:52 2011 from cf3840c2.endinfosys.com

```
* * * * * : W A R N I N G : * * * * *
*   THIS SYSTEM IS RESTRICTED TO AUTHORIZED EX LIBRIS SUPPORT STAFF   *
*   AND CUSTOMERS ONLY.                                             *
*   UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED AND MAY BE PUNISHABLE BY LAW. *
* * * * * : W A R N I N G : * * * * *
```

Other names for this server:

csm-voyager.hosted.exlibrisgroup.com.

Welcome to dc02vg0141na

[VGER] voyager@dc02vg0141na : voyager/ => █

# When You Login...

- A typical “terminal session” is when you login to gain access, execute commands to do work, then logout to terminate your connection.
- User → Shell → Kernel → Hardware/O.S.
- Command Syntax:
  - Command [-options] [arguments]
  - For example: `ls -l -a` (or `ls -la`)
  - For example: `tail -20 log.voyager`

# Three Important Things

---

- UNIX commands can't be abbreviated
- UNIX commands are case sensitive
- There are two metacharacters:
  - ? (match one character)
  - \* (match varied number of characters)

64.94.37.24 - PuTTY

```
$ find .??* -prune  
.TTauthority  
.Xauthority  
.bash_history  
.bash_logout  
.bash_profile  
.bashrc  
.dt  
.dtprofile  
.emacs  
.history  
.java  
.kde  
.lesshst  
.profile  
.profile-11-5-2011  
.profile.pre.7.0.1  
.sh_history  
.ssh  
.sunw  
.viminfo  
.zshrc  
$ █
```

# Talking to Your Server

- When you log in you enter commands at the command line or prompt:

```
$date Fri Mar 1 22:59:28 EDT 2011
```

```
$uptime 10:59PM up 259 days, 9:44, 5  
users, load average: 3.81, 14.27, 13.71
```

```
$hostname gonzo
```

- You can combine commands in a file called a **script** that allows you to run them one after another.

64.94.37.24 - PuTTY

```
/home/voyager => pwd
/home/voyager
/home/voyager => whoami
voyager
/home/voyager => PS1='$ '
$
$ PS1='$PWD => '
/home/voyager =>
/home/voyager => █
```

# Scripts are Cool

- Scripting allows you to automate tasks
- If you do it more than once, consider a script
- You can then use cron to schedule it to run automatically
- ExL recommends the Korn shell for scripting





# Example of Korn Shell Script

```
#!/bin/ksh  
date  
uptime  
hostname
```

To run: `./example.ksh`

Note the file must be **executable** (more later!)

# Who Am I?

- `whoami` – reports who YOU are logged in as
- `pwd` – reports current directory
- `who` – reports who is logged in
- `id` command – displays effective user and group identification
- `date` command – reports date and time



# Voyager versus Root User

- Most work is done as the voyager user
- The su and sudo commands let you become “other” users such as oracle or root:
  - `su - oracle`
  - `sudo -s`
- Always be aware of who you are!
- Always be aware of where you are!
- Be very careful when you're root

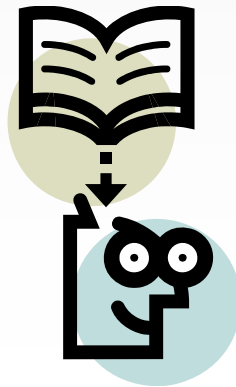
# Agenda

---

- What is an operating system?
- Logging in
- **The online manual**
- Directory basics
- File system hierarchy
- Users and groups
- File system commands
- The shell
- Process management
- Recap & Resources

# The Online Manual

- `man ls` (displays the `ls` man page)
- `man -k cat` (man entries with keyword `cat`)
- Entries can be very long!
- Special keys let you move about gracefully



# Contents of man pages

---

- Name
- Synopsis
- Description
- Return value
- Errors
- Bugs
- Warnings
- Examples
- Author
- Files
- See also

# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- **Directory basics**
- File system hierarchy
- Users and groups
- File system commands
- The shell
- Process management
- Recap & Resources

# Directory Basics

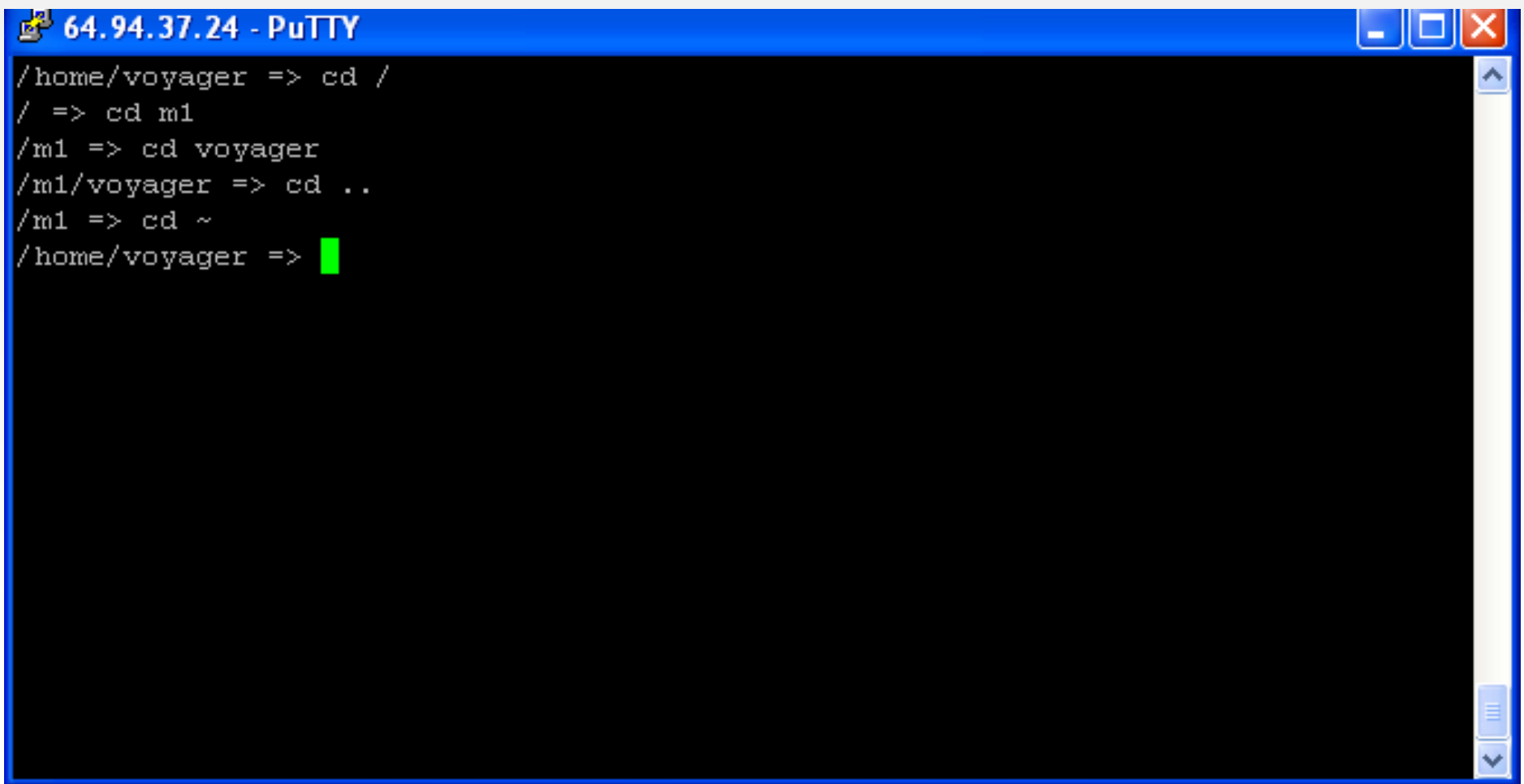
---

- UNIX organizes its files differently from Windows.
- If you're used to looking at Windows, at first the UNIX directory structure may seem illogical and strange, and you may have no idea where all the programs, icons, config files, and etc. are. Believe me! This is normal!



# Moving About

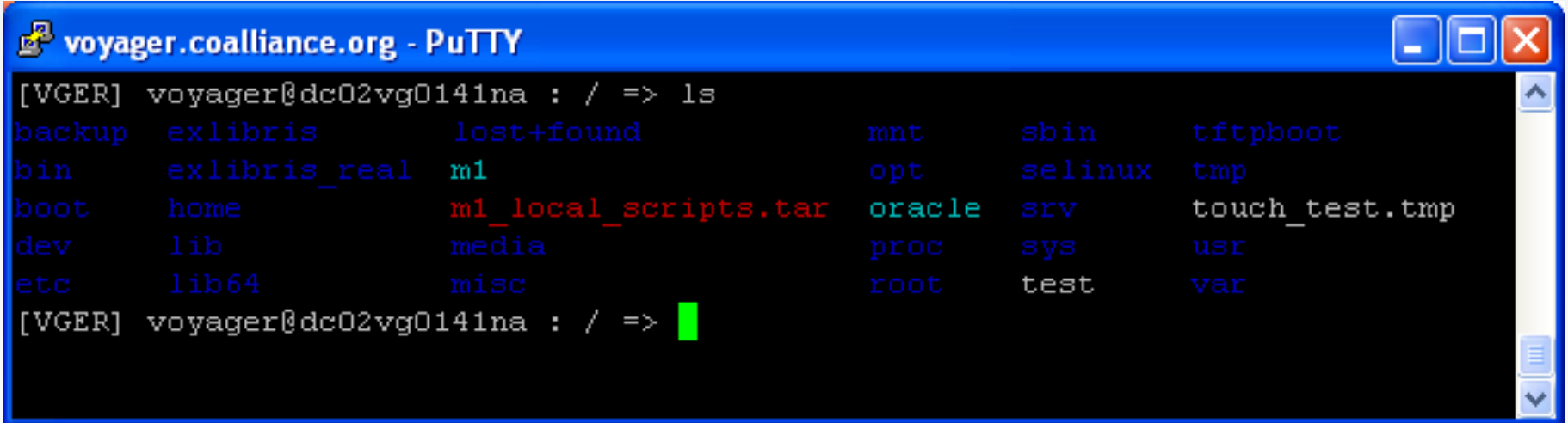
- The `cd` command allows you to change the directory you're in.
  - `cd subd` – moves you to the *subd* subdirectory
  - `cd /` – moves you to the root directory
  - `cd ..` – moves you to the next directory up
  - `cd ~` – moves you to your home directory
- The `ls` command lists the contents of a directory.



A screenshot of a PuTTY terminal window titled "64.94.37.24 - PuTTY". The window has a blue title bar with standard window controls (minimize, maximize, close) on the right. The terminal content is as follows:

```
/home/voyager => cd /  
/ => cd m1  
/m1 => cd voyager  
/m1/voyager => cd ..  
/m1 => cd ~  
/home/voyager => █
```

The terminal shows a sequence of directory changes: from the user's home directory to the root, then to a subdirectory 'm1', then to 'voyager' inside 'm1', then back to the parent directory, and finally back to the user's home directory. A green cursor is visible at the end of the last command.



The image shows a PuTTY terminal window titled "voyager.coalliance.org - PuTTY". The terminal displays the output of the 'ls' command executed by the user 'voyager' on the host 'dc02vg0141na'. The output lists various directories and files in a grid-like format. A green cursor is visible at the end of the prompt line.

```
[VGER] voyager@dc02vg0141na : / => ls
backup  exlibris      lost+found      mnt      sbin      tftpboot
bin     exlibris_real m1              opt      selinux   tmp
boot    home          m1_local_scripts.tar oracle    srv       touch_test.tmp
dev     lib           media          proc     sys       usr
etc     lib64        misc          root     test      var
[VGER] voyager@dc02vg0141na : / => █
```

# Common File Types

---

- Ordinary (regular or normal) Files
- Directory files
- Device files
- Symbolic links

# About Files

---

- Names are *case sensitive*
- Linux doesn't use file name extensions like Windows does
- There are more files on your system than you can see, because some of the files are *hidden files*
- Determine the type of a file using the `file` command

```
64.94.37.24 - PuTTY
/home/voyager => ls -la
total 75952
drwx----- 13 voyager    31016    4096 Jan 17 08:37 .
drwxr-xr-x 20 root      root      4096 Dec 29 15:51 ..
-rw----- 1 voyager    exlibris    76 Apr  2 2007 .TTauthority
-rw----- 1 voyager    exlibris   886 Nov 30 13:53 .Xauthority
-rw----- 1 voyager    exlibris  9293 Sep 28 15:26 .bash_history
-rw-r--r-- 1 voyager    31016     33 Sep 26 09:29 .bash_logout
-rw-r--r-- 1 voyager    31016    176 Sep 26 09:29 .bash_profile
-rw-r--r-- 1 voyager    31016    124 Sep 26 09:29 .bashrc
drwxr-xr-x 11 voyager    exlibris   4096 Apr  2 2007 .dt
-rwxr-xr-x 1 voyager    exlibris   5111 Aug  2 2006 .dtprofile
-rw-r--r-- 1 voyager    31016     515 Sep 26 09:29 .emacs
-rw----- 1 voyager    exlibris    0 Sep 27 08:58 .history
drwxr-xr-x 3 voyager    exlibris   4096 Apr 16 2008 .java
drwxr-xr-x 3 voyager    31016   4096 Sep 26 09:29 .kde
-rw----- 1 voyager    exlibris    54 Dec 19 13:42 .lesshst
-rw-r--r-- 1 voyager    exlibris  2151 Dec 19 07:51 .profile
-rw-r--r-- 1 voyager    exlibris  2135 Dec 19 07:49 .profile-11-5-2011
-rw-r--r-- 1 voyager    exlibris   907 Sep  5 2007 .profile.pre.7.0.1
-rw----- 1 voyager    exlibris 14598 Jan 27 10:11 .sh_history
drwx----- 2 voyager    exlibris   4096 Feb 19 2008 .ssh
drwx----- 3 voyager    exlibris   4096 May 22 2008 .sunw
-rw----- 1 voyager    exlibris  8916 Jan 17 08:37 .viminfo
-rw-r--r-- 1 voyager    31016    658 Sep 26 09:29 .zshrc
drwxr-xr-x 4 voyager    exlibris   4096 Jan 11 07:47 CLASS
drwx----- 2 voyager    exlibris   4096 Aug  2 2006 Mail
-rw-r--r-- 1 voyager    exlibris    0 Oct  8 2007 NOLOG
```

# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- Directory basics
- **File system hierarchy**
- Users and groups
- File system commands
- The shell
- Process management
- Recap & Resources

# File System Hierarchy

---

- /
- bin
- boot
- dev
- etc
- home
- lib



# File System Hierarchy

---

- **mnt**
- **sbin**
- **tmp**
- **usr**
- **var**

# The Voyager Server

- Extra “stuff” lives in root (“/”)
- /m1 – base directory for Voyager
- /m1/voyager – all Voyager files on the UNIX server are under this directory
- /m1/voyager/xxxdb – database directory containing all database-specific files



# Path Names

---

- A sequence of file names separated by slashes (/)
- **Absolute** pathname (start from the *root* directory)
  - For example: more /home/voyager/web.xml
- **Relative** pathname (start from the *current* directory)
  - more ./web.xml (.=current directory)
  - more ../ptrn-updt.Wed (..=parent directory)

64.94.37.24 - PuTTY

```
/home/voyager => cd /usr/X11R6/  
/usr/X11R6 => ls  
bin  
/usr/X11R6 => ls ./bin  
mkfontdir mkfontscale  
/usr/X11R6 => ls bin  
mkfontdir mkfontscale  
/usr/X11R6 => cd bin  
/usr/X11R6/bin => cd ..  
/usr/X11R6 => cd /usr/X11R6/bin/  
/usr/X11R6/bin => █
```

# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- Directory basics
- File system hierarchy
- **Users and groups**
- File system commands
- The shell
- Process management
- Recap & Resources

# Users and Groups

- Remember that UNIX is designed to be a multi-user environment. In an environment with more than one user, it is crucial to have a secure system for deciding which files are yours and who can fiddle with them.
- UNIX permissions are based on both users and groups



# File Ownership and Permissions

- Every file on a Linux system, including directories, is owned by a specific user and controlled by a group. Therefore, file permissions are defined separately for **users**, **groups**, and **others** (aka: everyone else or "world")

- Example:

```
/home/voyager/UNIX => ls -l /var/log/wtmp
```

```
-rw-rw-r-- 1 root utmp 97536 Jan 10 13:00 /var/log/wtmp
```

# A File's Permissions

- Consider the following example permissions, which have been color coded for convenience:

**-rw-rw-r--**



# More Permissions

- Consider the following example permissions, which have been color coded for convenience:

**-rw-rw-r--**

# In Practice

```
drwxr-xr-x 3 lguy class 80 2011-09-20 21:37 dir
```

```
-rw-r----- 1 lguy class 8187 2011-09-19 13:35 file
```

```
-rwxr-xr-x 1 lguy class 10348 2011-07-17 20:31 file2
```

# Changing Permissions

- If you are the owner of a file, you are allowed to change its permissions using the "chmod" command.
- The root user can change permissions for all files.
- chmod stands for "change mode", and "mode" is another word for permissions.

# chmod Syntax

- chmod's general syntax is:
- **chmod [who] + [permissions]**
  - to add permissions
- **chmod [who] - [permissions]**
  - to remove permissions
- **chmod [who] = [permissions]**
  - to replace permissions

64.94.37.24 - PuTTY

```
/home/voyager/CLASS/UNIX => ls -la testfile
-rw-r--r-- 1 voyager exlibris 21 Jan 10 13:20 testfile
/home/voyager/CLASS/UNIX => chmod a=r testfile
/home/voyager/CLASS/UNIX => ls -la testfile
-r--r--r-- 1 voyager exlibris 21 Jan 10 13:20 testfile
/home/voyager/CLASS/UNIX => chmod g+x testfile
/home/voyager/CLASS/UNIX => ls -la testfile
-r--r-xr-- 1 voyager exlibris 21 Jan 10 13:20 testfile
/home/voyager/CLASS/UNIX => chmod u+wx testfile
/home/voyager/CLASS/UNIX => ls -la testfile
-rwxr-xr-- 1 voyager exlibris 21 Jan 10 13:20 testfile
/home/voyager/CLASS/UNIX => chmod ug-x testfile
/home/voyager/CLASS/UNIX => ls -la testfile
-rw-r--r-- 1 voyager exlibris 21 Jan 10 13:20 testfile
/home/voyager/CLASS/UNIX => █
```

# chmod Syntax

- **chmod [who] + [permissions]**
- [who] can be any combination of **a** for **All** users, **u** for the **User** who owns the file, **g** for the **Group** who controls the file, or **o** for all **Other** users
- [permissions] can be any combination of **r** for **Read**, **w** for **Write**, and **x** for **eXecute**

# Our Script Example

- When you first create a shell script, it will usually not be executable. Use the **chmod** command with the `+x` option to add the execute permissions
  - `chmod a+x example.ksh`

64.94.37.24 - PuTTY

```
/home/voyager/CLASS/UNIX => ls -la example.ksh
-rw-r--r-- 1 voyager exlibris 33 Jan 10 13:02 example.ksh
/home/voyager/CLASS/UNIX => chmod a+x example.ksh
/home/voyager/CLASS/UNIX => ls -la example.ksh
-rwxr-xr-x 1 voyager exlibris 33 Jan 10 13:02 example.ksh
/home/voyager/CLASS/UNIX => ./example.ksh
Fri Jan 27 10:58:09 MST 2012
 10:58:09 up 10:52,  1 user,  load average: 0.00, 0.01, 0.00
dc02vg0179na.hosted.exlibrisgroup.com
/home/voyager/CLASS/UNIX => █
```



# Limitations

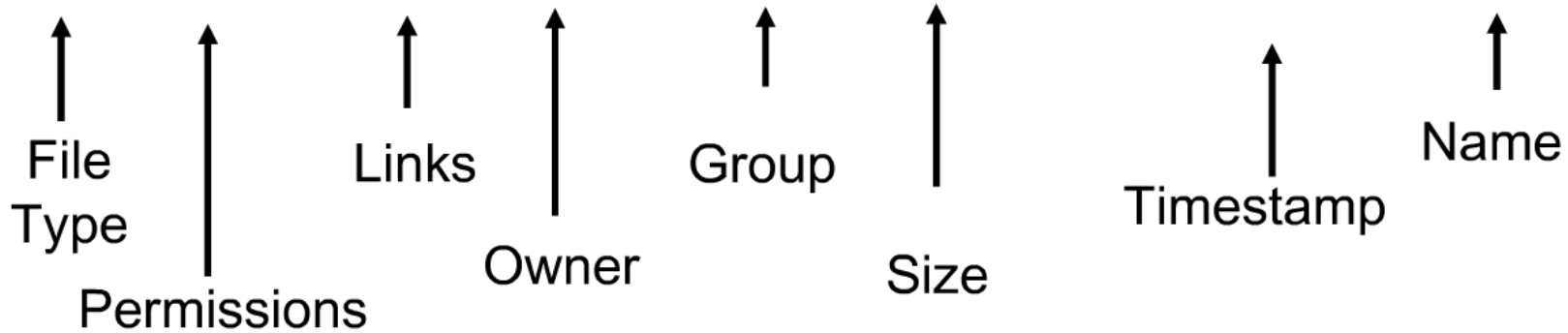
---

- Yes, user/group/other is all you get
- Difficult to do more complicated access lists
- Cannot express "this group of users can read, this group of users can write"
- There are complicated ways to get around this with setgid and nested directories
- The chown command changes file ownership

# File Characteristics

```
$ ls -l
```

```
-rw-r--r--  1 user3 class 37    Jul 24 11:06 f1  
-rwxr-xr-x  1 user3 class 52    Jul 24 11:08 f2  
drwxr-xr-x  2 user3 class 1024  Jul 24 12:03 memo
```



# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- Directory basics
- File system hierarchy
- Users and groups
- **File system commands**
- The shell
- Process management
- Recap & Resources

# File System Commands

---

- There are **two** kinds of commands: Shell Commands and Linux/Unix Commands.
- There are hundreds of Linux/UNIX commands
- But likely you will only need a small subset
- Remember that Linux/UNIX Commands are different than Shell Commands, but you get help for both in the man pages
- **File System Commands** are Linux/UNIX
- Don't forget your handouts!



The image shows a PuTTY terminal window titled "64.94.37.24 - PuTTY". The terminal output is as follows:

```
/home/voyager => echo $SHELL
/bin/ksh
/home/voyager => █
```

The terminal has a black background with white text. A green cursor is visible on the third line. The window title bar includes standard Windows-style minimize, maximize, and close buttons.

# pwd Command

- pwd – displays current directory:
  - `/home/voyager => pwd`
- To get brief help with the pwd type:
  - `/bin/pwd --help`
- Or view the man page:
  - `man pwd`

# ls Command

- ls – lists contents of current directory
  - `ls`
- ls filename – list a specific file
  - `ls my_file`
- ls /different/directory – lists contents of another directory
  - `ls /home/voyager`

# cd Command

- cd – change directory to subdirectory “hello”
  - cd hello
- cd / – change to root directory
  - cd /
- cd /m1/voyager – changes to voyager user home directory
  - cd /m1/voyager



# mkdir Command

---

- mkdir – creates a directory
  - `mkdir new_directory`

# cp Command

- cp – copies a file
  - `cp file new_file`
- cp – copies file to directory
  - `cp file dest_directory`
- cp -r – copies a directory recursively
  - `cp -r dir new_dir`

WARNING! THERE IS NO UNDO COMMAND!

# File Display Commands

- `cat` – display contents of a file
  - `cat file1`
- `wc` – counts lines/words/bytes in a file
- `less` – display contents of a file one screen at a time
  - `less log.voyager`
- You can open multiple files with `less`
  - `less file1 file2`

# tail Command

- tail – list last ten lines of a file
  - `tail filename1`
- tail -100 – list last 100 lines of a file
  - `tail -100 filename1`
- tail -100 | more – above but screen at a time
  - `tail -100 | more filename1`
- tail -f – output appended data as file grows
  - `tail -f log.voyager`

# rm Command

- rm – remove file
  - `rm filename1`
- rmdir – remove directory (if empty)
- rm -r – remove directory (if **not** empty)
  - `rm -r dir`

# mv Command

- mv – renames a file
  - `mv file new_file`
- mv – moves file to a directory
  - `mv file dest_directory`
- mv – rename or move directory
  - `mv dir new_dir`

# Other Useful Commands

---

- `clear` – clears all previously executed commands and their output from the current terminal
- `reset` – use the `reset` command if you've messed up your terminal
- `Ctrl + d` – logout and close your puTTY session

# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- Directory basics
- File system hierarchy
- Users and groups
- File system commands
- **The shell**
- Process management
- Recap & Resources



# The Shell

- The shell listens for commands and tries to execute them.
- Remember this?

User → Shell → Kernel → Hardware



# Possible Shell Functionalities

---

- Command execution
- Environment settings
- Variable assignment
- File name completion
- Command line recall and editing
- Input/Output redirection

# Commonly Used Shells

---

- /bin/bash – bash shell
- /bin/ksh – korn shell
- /bin/sh – bourne shell
- /bin/csh – C shell

The image shows a PuTTY terminal window titled "64.94.37.24 - PuTTY". The terminal output is as follows:

```
/home/voyager => echo $SHELL
/bin/ksh
/home/voyager => █
```

The terminal window has a blue title bar with standard window controls (minimize, maximize, close) on the right. The main area is black with white text. A green cursor is visible on the third line. On the right side of the terminal area, there are scroll-up and scroll-down arrows.

# Automatic File Name Completion

---

- Working at the Linux command line requires a lot of typing.
- It could get very frustrating with long commands and file names
- Luckily your Tab key can make your life a lot easier

# Command History

- The shell keeps a history of your commands
- The history command displays the last 16
- But it stores a lot more of them!
- You can recall, edit and re-enter commands
- `history -2` – list the last two commands
- `history 3 5` – list commands 3 through 5
- `r 1104` – run command number 1104
- `r echo` – run most recent starting w/ string

lguy@dc02vg0179na:/home/voyager/CLASS/UNIX

/home/voyager/CLASS/UNIX =>history

```
1330  ls -latr
1331  mkdir dir1
1332  cd dir1
1333  vi test
1334  ls
1335  cd ..
1336  ls
1337  cp dir1 dir2
1338  ls
1339  vi testfile2
1340  vi testfile2
1341  history
1342  history 3 5
1343  history 1000 1001
1344  clear
1345  history
```

/home/voyager/CLASS/UNIX =>history 1000 1005

```
1000  ps -ef | grep tomcat
1001
1002  ls
1003  more web.xml
1004  more ./web.xml
1005  ls
```

/home/voyager/CLASS/UNIX =>r 1000

ps -ef | grep tomcat

```
voyager  2677 13764  0 06:44 pts/0    00:00:00 grep tomcat
voyager  5546   1  0 00:06 ?          00:00:18 /m1/voyager/cominedb/tomcat/java/bin/java -serve
voyager  5825   1  0 00:06 ?          00:00:18 /m1/voyager/csm_traindb/tomcat/java/bin/java -se
voyager  6063   1  0 00:06 ?          00:01:24 /m1/voyager/csmdb/tomcat/java/bin/java -server -
voyager  6273   1  0 00:06 ?          00:00:18 /m1/voyager/csmphotosdb/tomcat/java/bin/java -se
voyager  6474   1  0 00:06 ?          00:00:19 /m1/voyager/ropewaydb/tomcat/java/bin/java -serv
voyager  6926   1  0 00:06 ?          00:00:19 /m1/voyager/tellertldb/tomcat/java/bin/java -se
```

/home/voyager/CLASS/UNIX =>

# The User Environment

- The env command
- Describes your session to programs you run
  - `/home/voyager => env`
  - `_=*12186*/bin/env`
  - `COLUMNS=182`
  - `CVS_RSH=ssh`
  - `DLIB=`
  - `EDITOR=vi`
  - `EPC_DISABLED=TRUE`
  - `G_BROKEN_FILENAMES=1`
  - `HISTSIZE=1000`
  - `HOME=/home/voyager`
  - ...



64.94.37.24 - PuTTY

```
/home/voyager => echo $SHELL
/bin/ksh
/home/voyager => SHELL=/bin/bash
/home/voyager => echo $SHELL
/bin/bash
/home/voyager => SHELL=/bin/ksh
/home/voyager => echo $SHELL
/bin/ksh
/home/voyager => █
```

# The PATH Variable

- The PATH variable is a list of directories where the shell will search for commands you type:
  - `PATH=/m1/utility/util:/m1/shared/bin:/bin:/usr/bin:/usr/ccs/bin:/etc:/usr/sbin:/usr/local/bin:/usr/openwin/bin:/sbin:/usr/sbin:/oracle/app/oracle/product/11.2.0/db_1/bin:/oracle/app/oracle/product/11.2.0/db_1/OPatch:.`



# Input/Output Redirection

- Redirect output TO a file:
  - `ls -la > list.txt`
- Redirect and append output TO a file:
  - `ls -la >> list.txt`
- Redirect input FROM a file:
  - `mail user < letter.txt`

# grep

- Search for lines matching specific pattern
- `grep [options] pattern [file1 file2]`
  - `grep error voyager.log`
- ignore the case of letters
  - `grep -i ERROR voyager.log`

64.94.37.24 - PuTTY

```
/m1/voyager/csmdb/log => pwd
/m1/voyager/csmdb/log
/m1/voyager/csmdb/log => grep -i error log.voyager | more
keysvr[1537] - ERROR - Thu Jun 19 09:59:00 2008
keysvr[1554] - ERROR - Thu Jun 19 09:59:01 2008
keysvr[1539] - ERROR - Thu Jun 19 10:00:35 2008
z3950svr[1872] - ERROR - Thu Jun 19 10:05:49 2008
z3950svr[1972] - ERROR - Thu Jun 19 10:05:50 2008
keysvr[2001] - ERROR - Thu Jun 19 10:05:51 2008
keysvr[2019] - ERROR - Thu Jun 19 10:05:51 2008
z3950svr[2121] - ERROR - Thu Jun 19 10:05:52 2008
z3950svr[2189] - ERROR - Thu Jun 19 10:05:53 2008
z3950svr[2281] - ERROR - Thu Jun 19 10:05:54 2008
z3950svr[2431] - ERROR - Thu Jun 19 10:05:56 2008
z3950svr[2589] - ERROR - Thu Jun 19 10:05:58 2008
z3950svr[2702] - ERROR - Thu Jun 19 10:05:59 2008
z3950svr[2881] - ERROR - Thu Jun 19 10:06:01 2008
z3950svr[2962] - ERROR - Thu Jun 19 10:06:02 2008
z3950svr[2998] - ERROR - Thu Jun 19 10:06:03 2008
z3950svr[3105] - ERROR - Thu Jun 19 10:06:03 2008
z3950svr[3159] - ERROR - Thu Jun 19 10:06:04 2008
```

# find

- Search one or more directory structures for files that meet certain criteria
- Search for files in whole system with name file1:
  - `find / -iname file1`
- You can find more than just file *names*!
- Use `-prune` test to stop from being recursive:
  - `find /foo -iname file* -prune`

64.94.37.24 - PuTTY

```
/m1/voyager/csmdb => find /m1/voyager/csmdb -name tsvrctl  
/m1/voyager/csmdb/tomcat/ews/tsvrctl  
/m1/voyager/csmdb/tomcat/tsvrctl  
/m1/voyager/csmdb => █
```

# diff

---

- Reports the differences between files
- Syntax: `diff [-options] file1 file2`
- Useful options include ignore white space and blank lines



64.94.37.24 - PuTTY

```
/home/voyager => cd CLASS  
/home/voyager/CLASS => cd UNIX  
/home/voyager/CLASS/UNIX => diff myscript example.ksh  
0a1  
> #!/bin/ksh  
/home/voyager/CLASS/UNIX => █
```

# tar

- Create a disk archive that contains a group of files or an entire directory structure:
  - `tar cvf panda.tar panda/`
- Or restore such an archive:
  - `tar -xvf panda.tar`

# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- Directory basics
- File system hierarchy
- Users and groups
- File system commands
- The shell
- **Process management**
- Recap & Resources

# Process Management

- ps command
  - `ps -ef` – Gives you list of all processes running including their unique process identification number (pipe it through the **less** command!)
- kill command
  - `kill 485` – Kill PID 485 *gracefully*
  - `kill -9 485` – *Strongly* kill PID 485

# Useful Voyager Process Commands

- `ps -ef | grep -i voyager`
- `ps -ef | grep -i ora_`
- `ps -ef | grep -i httpd`
- `ps -ef | grep -i java`
- `ps -ef | grep -i listener`
- `ps -ef | grep -i tomcat`

# Controlling CLI Processes

- Run Process in Foreground:
  - `ls -lR / > lsout`
  - `Control-c` to abort
- Run Process in Background:
  - `ls -lR / > lsout &`

# Running Long Processes

- The nohup command will stop a process from being killed if you log off before it finishes
  - `nohup ls -lR / >lsout &`

# Job Control

control-z	Stop (don't kill) the foreground job, and then return to the shell
jobs	Check the status of jobs in the current session
kill -9 %1	Kill a <i>job</i> , by specifying its job number after the percent sign
kill -9 123	Kill a <i>process</i> , by specifying its process id (PID) number
bg	Run the most recently stopped job in the background
fg	Bring most recently backgrounded job to the foreground
fg %1	Bring a job to foreground by specifying its job number after the percent sign



# Cron

---

- cron is a UNIX utility that allows tasks (such as running a script!) to be automatically run in the background at regular intervals by the cron daemon. These tasks are often termed as cron jobs.
- crontab (CRON TABLE) is a file which contains the schedule of cron entries to be run and at specified times.

# Crontab Commands

- `crontab -e` Edit your crontab file, or create one if it doesn't already exist.
- `crontab -l` Display your crontab file.
- `crontab -r` Remove your crontab file.
- **Note** : If you inadvertently enter the crontab command with no argument(s), do not attempt to get out with Control-d. This **removes all entries** in your crontab file. Instead, exit with Control-c.

# Agenda

---

- What is an operating system?
- Logging in
- The online manual
- Directory basics
- File system hierarchy
- Users and groups
- File system commands
- The shell
- Process management
- **Recap & Resources**

# Recap & Resources

---

- PuTTY <http://www.putty.org/>
- Discussion list for Ex Libris' Voyager Integrated Library System [VOYAGER-L@listserv.nd.edu](mailto:VOYAGER-L@listserv.nd.edu)
- [voyager-administrators@googlegroups.com](mailto:voyager-administrators@googlegroups.com)
- The Internet
- Books, books and more books
- The UNIX man pages, apropos command, etc.