Directories and files

Concepts

- ❖ Files are organised in a directory tree/hierarchy
- Everything is a file (e.g. keyboard, printers, ...)
- * Each process has access to the files stdin (input), stdout (buffered output), stderr (unbuffered output)
- Each process operates in a working directory
- Each user has a home directory

Paths

Path = Identifier for the location of file/directory

- Paths consists of a parent directory list + file/directory
- ❖ Files and directories are separated by a '/'
- Directory paths may contain a trailing '/'

Absolute path = Full location (first character = '/')

Relative path = Relative location (first character \neq '/')

Path to the directory itself Path to the parent directory

/usr/bin/ls Example for an absolute file path Example for an absolute directory path /home/foo/

Example for a relative file path ./a.out

File system hierarchy

Root directory /

/bin Essential command executables

/dev Device files

System-wide configuration files /etc

Manually added software /opt

/sbin Essential administrative executables Temporary files /tmp

System resources for users /usr

Command executables /usr/bin

/usr/local Site-local data

Administrative executables /usr/sbin

/var Variable files

man hier (or man file-hierarchy on recent Linux Expansions distributions) to get a more detailed overview

Terminal (emulator)

Text terminal = Computer interface for text entry/display Terminal emulator = Application that emulates a text termi*nal* in a graphical environment

Examples for terminal emulators: xterm, urxvt, quake

Opening a terminal

Ctrl + Alt + T Unity/GNOME $\mathsf{Cmd} + \mathsf{Deg} \to \mathsf{"terminal"} \to \mathsf{Deg}$ Mac OS $[\mathsf{Win}] + [\mathsf{R}] o \mathsf{"bash"} o [\downarrow]$ Bash on Windows

Shell

Unix shell = User interface that accepts commands to operate a computer

man intro to get an introduction into basic shell usage

Examples for shell programs: sh, bash, zsh, fish, ksh

Prompt

Prompt = Text sequence that precedes each line that prompts the user to enter a command

[foo@bar /var/www]\$ Example prompt in bash \Rightarrow User foo is operating in the working directory / Var/www at the computer with the host name bar

Line editing



Go to beginning/end of the line Erase to beginning/end of the line Go one word forward/backward in line Erase to beginning of current word Cancel the current command line

Metacharacters

The following characters have special meaning and sometimes they can't be used directly as arguments/words:

Their special meaning can be disabled:

Preserves the literal value of the following character

' ' Preserves the literal values of enquoted characters

Like ' ' but characters ` \$ \ retain their meaning

~	Home directory of the current user
*	Matches any character sequence
?	Matches a single character
[]	Matches any one of the enclosed characters
\${ <i>var</i> }	Value of the environment variable <i>var</i>
\$(cmd)	Output of <i>cmd</i>
\$((<i>expr</i>))	Result of the mathematical expression expr

Shell utilities anrenes toyt

apropos text cat file cd dir chmod prm file cp src dst echo text exit file file find dir expr grep expr file ls dir man cmd mkdir dir mv src dst pwd	Prints the contents of file Changes the working directory to dir Changes permissions of file to prm Copies the file/directory src to dst Prints text Exits the shell Determines the file type of file Finds files in dir that match expr Searches for pattern expr in file List the entries in the directory dir Displays the manual for cmd Creates the directory dir Moves/renames src to dst Prints the current working directory
rm file touch file	Removes the file <i>file</i> Creates the empty file <i>file</i>
	' '

Input output redirection

cmd1 cmd2	Runs cmd1 and cmd2 and redirects the
	output of <i>cmd1</i> to the input of <i>cmd2</i>
<pre>cmd > file</pre>	Runs <i>cmd</i> and redirects output to <i>file</i> ,
	content of <i>file</i> is overwritten
<pre>cmd >> file</pre>	Like >> but appends output to file
<pre>cmd < file</pre>	Runs <i>cmd</i> and redirects <i>file</i> to its input
<pre>cmd <<< text</pre>	Runs <i>cmd</i> with input <i>text</i>

Job control

Ctrl + D

Job = Shell command and its associated process(es)

- * Each job has a job id and corresponding process ids
- ❖ Jobs can run in the foreground or in the background
- ❖ The execution of a job can be temporarily suspended

Starts <i>cmd</i> as background job (id is printed)
Puts job with id jid in foreground
Continues job with id jid in background
Prints ids of jobs in current shell
Terminates process with the PID <i>pid</i>
Prints PIDs of processes in current terminal
Suspends/resumes active job
Puts job to background and suspends it
Aborts job (most of the times)

Sends an EOF character