

Task Optimization



Mateo Prigl
SOFTWARE DEVELOPER



Demo



Brace expansion



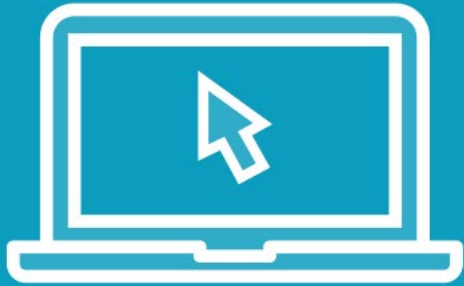
Demo



Build commands from stdin with xargs



Demo



Checking for destructive operations



Demo



Opening files in external programs



Summary



Brace expansion

Build commands with xargs

Checking for destructive operations

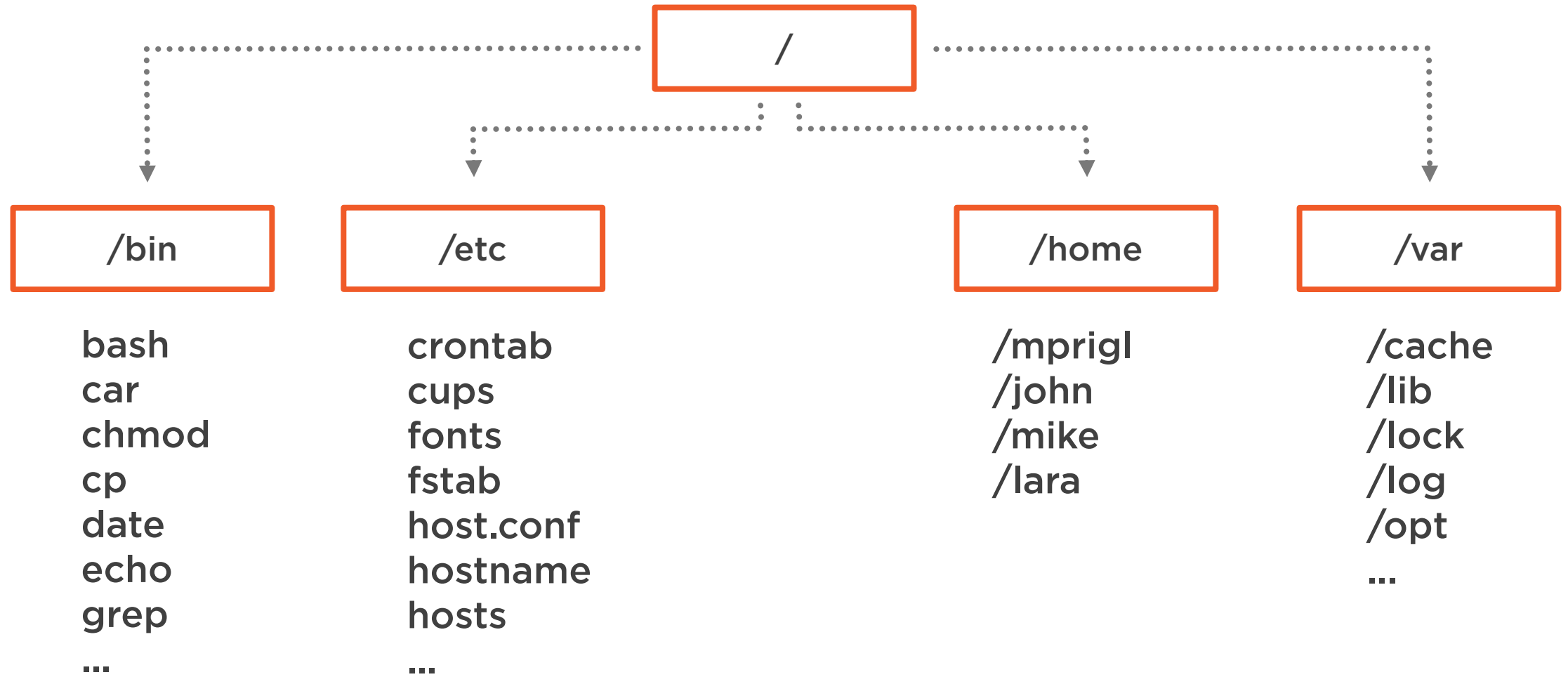
Opening files in external programs



Summary



UNIX Filesystem



Navigation in Shell

~ sign represents your home directory
Check your current directory with pwd
Move to other directories with cd



File Types



Directories – contain other files



Regular files



Globbing

Match patterns with wildcards

* matches zero or more characters

? matches exactly one character

[A-Z] will match any letter of the alphabet

Z Shell has glob qualifiers



Managing Files

Use find to locate files by name

Create files with touch

Create directories with mkdir

Remove empty directories with rmdir

Remove files with rm

Remove non-empty directories with rm -r



Command-line Text Editors

nano

vi

emacs



Reading File Contents

Page through the file with more and less

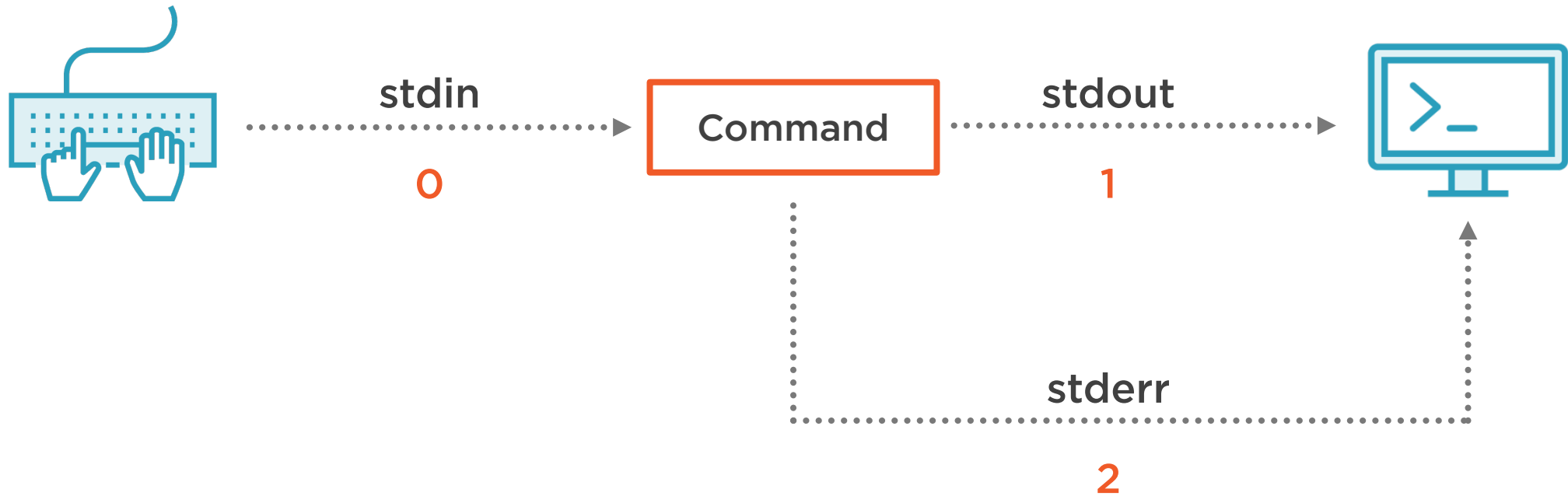
Read the first lines of a file with head

Read the last lines of a file with tail

Count the number of lines with wc



Input/Output Redirection



Input/Output Redirection

Redirect standard input with <

Redirect standard output with >

Redirect standard error with 2>

Create command pipelines with |



Processing Lines

Pick the data from a line with cut

Translate characters with tr

Get unique lines with uniq

Sort lines with sort



File Permissions

Permissions

File

Directory

Read (r)	Open and read	List the contents
Write (w)	Edit (modify) contents	Add, remove and rename files within it
Execute (x)	Execute (executable file)	Enter inside (cd into it)



Changing Permissions

File Permissions (symbolic mode)

chmod

u user

g group

o other

a all

=

set permissions

+

add permission

-

remove permission

r w x

filename



Changing Permissions

File Permissions (absolute/numeric mode)

chmod

	4	2	1	
	r	w	x	
1	0	0	1	- - x
2	0	1	0	- w -
4	1	0	0	r - -
1+2=3	0	1	1	- w x

filename



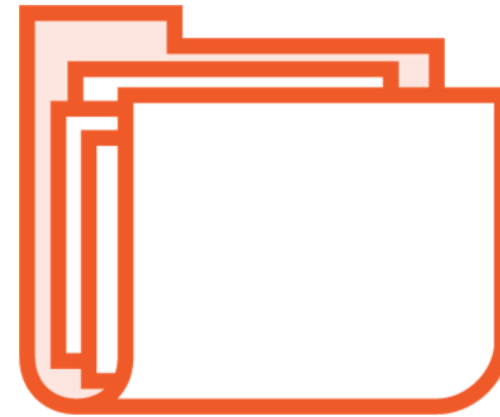
Default UNIX Permissions



File Permission

666

rw-rw-rw-



Directory Permission

777

rwxrwxrwx



Default File Permissions

Default
permissions

6

6

6

rw-rw-rw-

Umask

-

0

2

2

--- -w- -w-

6

4

4

rw- r-- r--



Managing Users and Groups

Create users with `useradd`

Set the login password with `passwd`

User information is stored in `/etc/passwd`

Create groups with `addgroup`

Login to other account with `su - username`



Special File Permissions

Permissions	File	Directory
SUID (s or S)	Run file as a file owner	
SGID (s or S)	Run file as a member of the file group	All of the files inside of it will belong to the directory group
Sticky bit (t or T)		Prevents files inside of it from being removed by anyone but the owner of that file



Moving Files

Copy files with cp

Move and rename files with mv

Backup and synchronize files with rsync



Links

Soft (symbolic) links only hold the filename of a file they point to

Hard links point to inodes



Processing Text

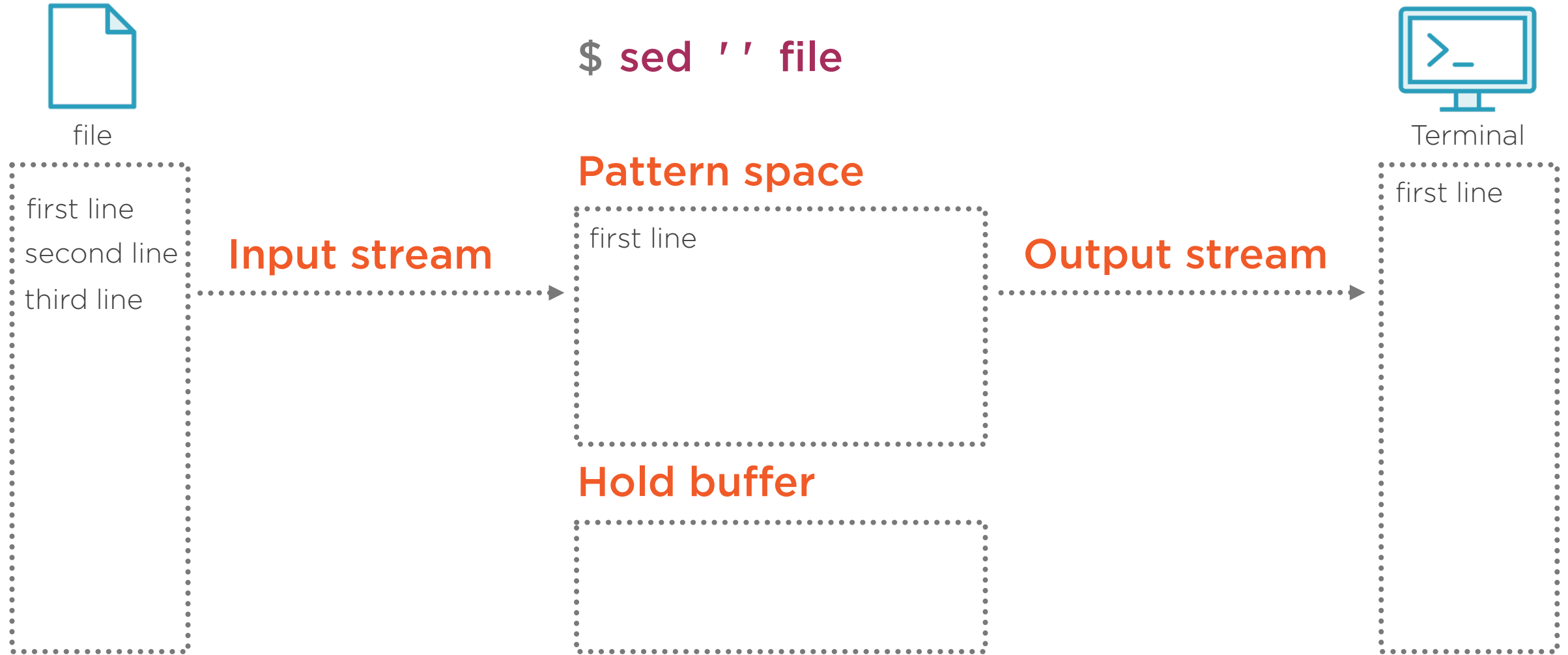
Search for patterns with grep

Transform text with sed and awk



SED Workflow

\$ sed ' ' file



AWK Workflow

awk

```
'BEGIN {  
    print "execute before everything"  
}  
/expression/ {  
    print "execute on each line from the input stream"  
}  
END {  
    print "execute after everything"  
}'
```

filename



Tar Options

Option

Purpose

v	Verbose output
f	Use an archive file
c	Create an archive
t	List the archive content
x	Extract files from an archive
z	Gzip compression
j	Bzip2 compression

