

Input-Output Redirections

File Descriptors		
N°	Name	Generally
0	Standard Input	Keyboard
1	Standard Output	Screen
2	Standard Error Output	Screen

Output Redirection

- **Overwrite or Create file:** `input > output`

Sample:

```
$ cat /etc/hosts > /tmp/hosts
$ ls /home > /tmp/userslist
```

- **Append or Create:** `input >> output`

Sample:

```
$ echo "192.168.1.1 gateway" >> /etc/hosts
$ echo "myservice 1099/TCP" >> /etc/services
```

Input Redirection

- **Use file as input:** `command < input`

Sample:

```
# Sort the services listed on /etc/services by name
$ sort < /etc/services
```

- **Here document:** `<<DELIMITER`

Sample:

```
$cat << EOF
NEW FILE
EOF
```

Error Redirection

- **Redirect errors to a file:** `command 2> output`

Sample:

```
#Listing a non existing file will gives you error
$ ls /NonExistingFile 2>/tmp/notfound
#Listing / as normal user will gives you access errors
$ ls -lR /root 2>/dev/null
```

Redirection Combinations

Input Redirection + Output redirection:

```
$ grep user < /etc/passwd > /tmp/userid-found
```

Here document redirect (Input+Output):

```
$ cat << DELIMITER >> /tmp/newfile
it is a new file
DELIMITER
```

Send both error and standard output to the same file:

```
$ find / -name "some.file" >/tmp/output 2>&1
```

Send error and output to different files:

```
$ grep 192.168.1.1 /etc/* 2>/tmp/errors >/tmp/found
```

Other odd Input/Output redirections

Create a empty file

```
$ > new-empty-file
```

Load a file into a variable

```
$ variable=$(</etc/hosts)
```

Resume table

Command	Description
<code>cmd > file</code>	Output of cmd is redirected to file
<code>cmd < file</code>	Program cmd reads its input from file.
<code>cmd >> file</code>	Output of cmd is appended to file.
<code>n > file</code>	Output from stream with descriptor n redirected to file.
<code>n >> file</code>	Output from stream with descriptor n appended to file.
<code>n >& m</code>	Merge output from stream n with stream m.
<code>n <& m</code>	Merge input from stream n with stream m.
<code><< delimiter</code>	Standard input comes from here through next delimiter at start of line

Pipes

Common use:

```
command1 | command2
```

Samples:

```
# Count lines in a file
$ cat /etc/hosts | wc -l
24
# Search a string in a output of a command
$ ip addr show | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
```

Grep basic RegExp

```
^ Start with
$ End with
\ Turn off the special meaning of the next character, as in \\.
[] Match any one of the enclosed characters, as in [aeiou].
Use Hyphen "-" for a range, as in [0-9].
[^] Match any one character except those enclosed in [ ], as in [^0-9].
. Match a single character of any value, except end of line.
* Match zero or more of the preceding character or expression.
\{x,y\} Match x to y occurrences of the preceding.
\{x\} Match exactly x occurrences of the preceding.
\{x,\} Match x or more occurrences of the preceding.
```

Samples:

```
# Grep a line starting with 127
$ grep "^127" /etc/hosts
127.0.0.1 localhost
# Grep a line ending with vm
$ grep "vm$" /etc/hosts
192.168.1.9 hostvm
```

Compression and Packaging

Tar

```
Create tar file: tar cvf /tmp/backup.tar /home/user/*
List files on a tar: tar tvf /tmp/backup.tar
Create a tar.gz file: tar zcvf /tmp/compressed-bkp.tar.gz /home/user/*
List files on a tar.gz: tar tzvf /tmp/compressed-bkp.tar.gz
Create a tar.bz file: tar jcvf /tmp/compressed-bkp.tar.bz /home/user/*
List files on a tar.bz: tar jtvmf /tmp/compressed-bkp.tar.bz
UnTar: tar xvf /tmp/backup.tar
UnTar and decompress tar.gz: tar zxvf /tmp/compressed-bkp.tar.gz
UnTar and decompress tar.bz: tar jxvf /tmp/compressed-bkp.tar.bz
```

```
--selinux Archive the SELinux attributes of the files and directories
--acls Archive the ACL attributes of files and directories
--xattrs Archive all Extended Attributes of files and directories.
```

Gzip and Bzip2

```
Compress gzip: gzip file
Compress bzip: bzip2 file
```

```
Decompress: gunzip file.gz or gzip -d file.gz
Decompress: bunzip2 file.bz or bzip -d file.bz
```

Start

start is not installed by default (yum install star)

Create star file raining the extended attributes:

```
$ star -xattr -H=exustar -c -f=/tmp/file.star /home/user/*
```

Extract star file:

```
$ star -x -f=/tmp/file.star
```

Hard and soft links

Hard link:

- can't be done accross mount points
- have the same inode

```
$ ln /etc/hosts /root/hosts
```

Soft link

- can be done accross mount points
- different inode

```
$ ln -s /etc/hosts /tmp/hostsfile
```

```
$ cp -s /etc/hosts /tmp/hostsfile
```

User and Group file perms

```
chmod [ugo|a] [+|-|=] [rwxsut]
```

u	User access
g	Group access
o	Other system user's access
a	Equivilent to "ugo"

r	Permission to read a file Permission to read a directory (also requires "x")
w	Permission to delete or modify a file Permission to delete or modify files in a directory
x	Permission to execute a file/script Permission to read a directory (also requires "r")
s	Set user or group ID on execution.
u	Permissions granted to the user who owns the file
t	Set "sticky bit. Execute file/script as user root for regular user.

+	Add access
-	Remove access
=	Access explicitly assigned

Symbolic Notation	Octal Notation	English
-----	0000	no permissions
---x--x--x	0111	execute
--w--w--w-	0222	write
--wx-wx-wx	0333	write & execute
-r--r--r--	0444	read
-r-xr-xr-x	0555	read & execute
-rw-rw-rw-	0666	read & write
-rwxrwxrwx	0777	read, write, & execute

File management

Create file: touch file **or** >file

Copy file: cp file file2

Move file: mv file file2

Create dir: mkdir folder

Copy dir: cp -r folder folder2

Move dir: mv folder2 folder3

Delete file: rm file

Delete dir: rm -r folder

Offline documentation

Command help (usually): command -h || command --help

Search command on manuals: apropos command

Manual title of command: whatis command

Command manual: man command || man x command

* replace x for manual page