Manipulate files and data

OS Practice CSC 420 Spring 2022

README

- Before working through this file, make sure you completed the getting started guide in `start_nb.org`.
- You can hide/open headers and codeblocks with the <TAB> key
- You can get this file and the `.emacs` file from GDrive
- The solutions and results herein were all obtained on a Pi

Building a playground

- Change to your `$HOME` directory
- Print your working directory
- Make a directory `playground`
- Switch on the verbose option
- Check that it worked using the wildcard `play*`

```
cd $HOME
pwd
mkdir -v playground
echo "found:" play*
```

```
/home/pi
mkdir: created directory 'playground'
found: playground
```

- `[X]` Check task group

Creating directories

- Change directory to `playground`[1] - do this in every code block!
- Print your working directory
- Make two directories `dir1` and `dir2`
- Switch on the verbose option
- Check that it worked using the wildcard `*[0-9]`

```
cd ~/playground
pwd
mkdir -v dir1 dir2
echo "found:" *[0-9]
```

```
/home/pi/playground
mkdir: created directory 'dir1'
mkdir: created directory 'dir2'
found: dir1 dir2
```

- `[X]` Check task group

Copying files

- Copy `/etc/passwd` into the current working directory
- Switch on the verbose option
- Check that it worked using the wildcard `pass??`

```
cd ~/playground
cp -v /etc/passwd .
echo "found:" pass??
```

```
'/etc/passwd' -> './passwd'
found: passwd
```

- `[X]` Check task group

Moving and renaming files

- Change the name of `passwd` to `fun`
- Switch on the verbose option
- Check that it worked with the wildcard `*fun*`

```
cd ~/playground
mv -v passwd fun
echo "found:" *fun*
```

```
renamed 'passwd' -> 'fun'
found: fun
```

- Move the renamed file `fun` to directory `dir1`
- Check that it worked with `ls -l`
- Move `fun` from `dir1 to ~dir2` in one command
- Check that it worked with `ls -l`
- Move `fun` back to the current working directory
- Check that it worked with `ls -l`

```
cd ~/playground
mv -v fun dir1
ls -l dir1
mv -v dir1/fun dir2
ls -l dir2
mv -v dir2/fun .
ls -l .
```

```
renamed 'fun' -> 'dir1/fun'
total 4
-rw-r--r-- 1 pi pi 1992 Mar 10 11:52 fun
renamed 'dir1/fun' -> 'dir2/fun'
total 4
-rw-r--r-- 1 pi pi 1992 Mar 10 11:52 fun
renamed 'dir2/fun' -> './fun'
total 12
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir1
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir2
-rw-r--r-- 1 pi pi 1992 Mar 10 11:52 fun
```

- Move file `fun` into `dir1` again
- Move directory `dir1` into `dir2`
- Confirm that the file is there with `ls -l`

```
cd ~/playground
mv -v fun dir1
mv -v dir1 dir2
ls -l dir2/dir1
```

```
renamed 'fun' -> 'dir1/fun'
renamed 'dir1' -> 'dir2/dir1'
total 4
-rw-r--r-- 1 pi pi 1992 Mar 10 11:52 fun
```

- Note that `dir1` was moved into `dir2` because it existed
- If it had not existed, `dir1` would have been renamed `dir2`
- Put everthing back and confirm at the end with `ls -l`

```
cd ~/playground
mv -v dir2/dir1 .
mv -v dir1/fun .
ls -l ~/playground
```

```
renamed 'dir2/dir1' -> './dir1'
renamed 'dir1/fun' -> './fun'
total 12
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir1
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir2
-rw-r--r-- 1 pi pi 1992 Mar 10 11:52 fun
```

- `[X]` Check task group

Creating hard links

- Create a hard link `fun-hard` to `fun` in `./`
- Create a hard link `fun-hard` to `fun` in `dir1`
- Create a hard link `fun-hard` to `fun` in `dir2`
- Switch on the verbose option for `ln`
- Confirm with `ls -l ./` and with `ls -l dir*`

```
cd ~/playground
ln -v fun fun-hard
ln -v fun dir1/fun-hard
ln -v fun dir2/fun-hard
ls -l .
ls -l dir*
```

```
'fun-hard' => 'fun'
'dir1/fun-hard' => 'fun'
'dir2/fun-hard' => 'fun'
total 16
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir1
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir2
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun
```

```
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
dir1:
total 4
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard

dir2:
total 4
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
```

- The number `4` in the listing is the number of hard links that exist for the file (including the default link)
- Show that `fun` and `fun-hard` are indeed the same files

```
cd ~/playground
ls -li fun*
```

```
272668 -rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun
272668 -rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
```

- `[X]` Check task group

Creating symbolic links

- Create a symlink `fun-sym` to `fun` in `./`
- Create a symlink `fun-sym` to `fun` in `dir1`
- Create a symlink `fun-sym` to `fun` in `dir2`
- Switch on the verbose option for `ln`
- Confirm with `ls -l ./` and with `ls -l dir*`

```
cd ~/playground
ln -vs fun fun-sym
ln -vs fun dir1/fun-sym
ln -vs fun dir2/fun-sym
ls -l .
ls -l dir*
```

```
'fun-sym' -> 'fun'
'dir1/fun-sym' -> 'fun'
'dir2/fun-sym' -> 'fun'
total 16
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir1
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir2
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
lrwxrwxrwx 1 pi pi    3 Mar 10 11:52 fun-sym -> fun
dir1:
total 4
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
lrwxrwxrwx 1 pi pi    3 Mar 10 11:52 fun-sym -> fun

dir2:
total 4
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
lrwxrwxrwx 1 pi pi    3 Mar 10 11:52 fun-sym -> fun
```

- Create a symlink `dir1-sym` to `dir1` in `./`

```
cd ~/playground
ln -vs dir1 dir1-sym
ls -l ./dir1*
```

```
'dir1-sym' -> 'dir1'
lrwxrwxrwx 1 pi pi    4 Mar 10 11:52 ./dir1-sym -> dir1

./dir1:
total 4
-rw-r--r-- 4 pi pi 1992 Mar 10 11:52 fun-hard
lrwxrwxrwx 1 pi pi    3 Mar 10 11:52 fun-sym -> fun
```

- `[X]` Check task group

Removing files and directories

- Remove the hard link `fun-hard` in `./` (with verbose option)
- Confirm with `ls -l`

```
cd ~/playground
rm -v fun-hard
```

```
removed 'fun-hard'
```

- Remove `fun` and switch on verbose option[2]
- Confirm with `ls -l`

bash

```
cd ~/playground
rm -iv fun
ls -l
```

```
rm: remove regular file 'fun'? removed 'fun'
total 8
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir1
lrwxrwxrwx 1 pi pi    4 Mar 10 11:52 dir1-sym -> dir1
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir2
lrwxrwxrwx 1 pi pi    3 Mar 10 11:52 fun-sym -> fun
```

- `[ ]`

  Check that `fun-sym` is broken now with `less` (don't do this in Emacs, but change to a proper terminal instead). You should get:

  ```
  fun-sym: No such file or directory
  ```

- `[ ]` Make sure that you understand what "broken symbolic link" in this context means, and why `fun-sym` is now broken
- Remove the symbolic links (switch on verbose option)
- Confirm with `ls -l`

```
cd ~/playground
rm -v fun-sym dir1-sym
ls -l
```

```
removed 'fun-sym'
removed 'dir1-sym'
total 8
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir1
drwxr-xr-x 2 pi pi 4096 Mar 10 11:52 dir2
```

- Go `$HOME` and remove the playground (with verbose option)
- Check with `ls -vl`

```
cd ~/
rm -vr playground
ls -vl
```

```
removed 'playground/dir2/fun-hard'
removed 'playground/dir2/fun-sym'
removed directory 'playground/dir2'
removed 'playground/dir1/fun-hard'
removed 'playground/dir1/fun-sym'
removed directory 'playground/dir1'
removed directory 'playground'
total 40
drwxr-xr-x 2 pi pi 4096 Jan 27 19:14 Bookshelf
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Desktop
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Documents
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Downloads
drwxr-xr-x 9 pi pi 4096 Mar  8 15:13 GitHub
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Music
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Pictures
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Public
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Templates
drwxr-xr-x 2 pi pi 4096 Jan 27 19:31 Videos
```

- `[ ]` Check this last task group
- Save this file with `C-x C-s`
- Kill the buffer with `C-x k` (confirm)

You may close Emacs!

Command summary

- `[ ]` Complete the table!

COMMAND	MEANING	EXAMPLE
cd		
pwd		
mkdir		
echo		
mv -v		

COMMAND	MEANING	EXAMPLE
rm -vr		
ln -vs		
ls -l		

Footnotes:

[1] If you work with code blocks inside Emacs, you may have to resort to absolute filenames to make sure that you are where you want to be.

[2] In Org-mode, you need to use the `:cmdline` header argument and redirect the input, in this case from a file `y` that only contains the character `y`, which I created for this purpose.

Author: Marcus Birkenkrahe
Created: 2022-03-10 Thu 11:52
Validate