Wenfeng Zhong
Dr. Brian
CS 164
19 Oct 2019
Fall 2019

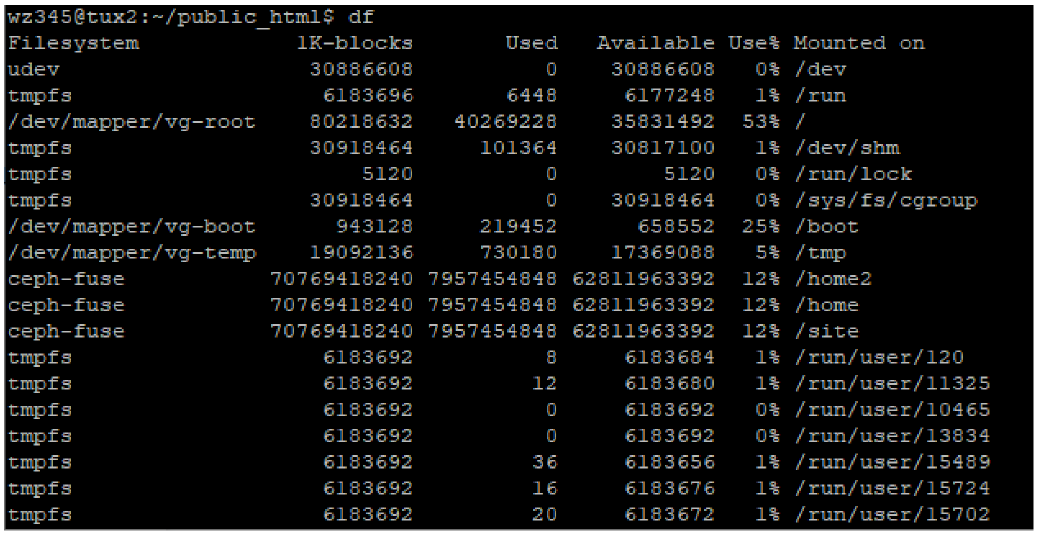
Unix File System

Unix is one of if not the most important operating systems in history. The smartphones we use today and even the most advanced modern operating systems all have their roots in Unix. Unfortunately, the birth of Unix was not as smooth as what people think. Many people thought that Unix was developed by a bunch of programmers with highly advanced technologies of its time to help make the operating system just like how companies develop modern systems today. But this is not the case for Unix, in fact, Unix wasn't even created intentionally. It was not until the end of the cooperative project "Multics" which inspired Ken Thompson about the concept of time sharing on an operating system and the game experiment and file system that he made using the PDP-7. The interesting thing about Unix definitely has the be the file system, it is the heart of Unix and it is what made Unix so famous. Even in the present day, the Unix file system is still being used, like the Tux server in Drexel.

In 1969, after the cancellation of the Multics time sharing project, Bell Labs lost interest in researching operating systems. The creator of Unix Thompson and his team however still believes that the time sharing concept that was intended to be used on Multics was more than using convenience, it also encourages close communication. So then a team consist of Thompson, Ritchie and Canady was formed. At the start, they were sketching the design of a file system on paper, but at the end, they were able to find a rarely used PDP-7 and used it to test the file system. Which, at the end Unix was born .

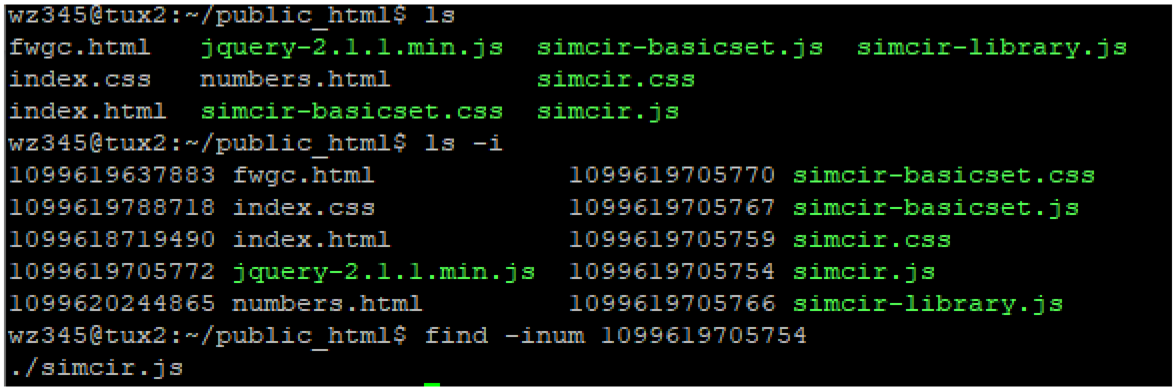
The structure of Unix is unique, the file system operates in a tree-like structure with the system file"root" at the top and the user files at the bottom. Directory is a type of special file that has other files in it. Within the "root" file, there are files that contains system operations like "usr", which contains the executable of Unix, the "temp" where all the temporary files are stored and also the user directory "home", the starting point when the user opens a shell. Unix treats everything as a file, even with devices like hard disks and scanners and the system will create a special file to map those devices. The whole system of Unix runs on commands that the users send in the shell. Commands like "ls" which list the files within the directory and "cp" which copy the files in the directory are some of the most common commands in Unix. Structurally the file system on the PDP7- version Unix was similar yet different to the modern day Unix. It was similar because It had an "i-list", Directories and special files. Also The basic system calls like "start, Read, write, open, create and close" were there at the start ("The Evolution of the Unix Time-Sharing System*." *Early Unix History and Evolution*). It was different because there was no ease and kill process and since the PDP-7 is a word addressing machine, the unit for I/O was word instead of bytes. However the most significant difference was that there were no path names and the file name shows on the terminal was just the name relative to the current directory. But the command "link" which is basically a shortcut to a directory or file did exist, and with a set of conventions, it became the alternative for path name.

The "i-list" is a special and important thing for Unix, and it essentially points to a special storage area that contains the operating system called root file system. It was created because it allows the system to map the memory in physical storage devices so it knows where the files are located, it also made commands like "cd" and "ls" possible. Since files are often deleted and added, there needs to have something that is flexible enough to track the files. The "i-node" is a part of the "i-list" and it provides the flexibility to track down files in the physical storage device. It also contains important information about the files, so much so that it would usually take up 1% of the total storage. ("It Is All about the Inode." *IBM Developer*) The amounts of inode is important, and to check the total number of inode in the file, you can use the command "df" and it will display the inodes status in the file system. In which 3rd and 4th column shows the used and available inodes in the file system and the last column showing where the removable files systems are mounted on the main file system. Unlike other operating systems where you would not be able to create new files only when the disk is filled, Unix would not allow you to create new files when it ran out of inodes, even when there are plenty of space left in the disk, and so tracking the numbers of inode in the system is an important part in Unix. ("It Is All about the Inode." *IBM Developer*) In the topic of tracking, there are numbers called inode numbers that refers to each inodes, by using the command "ls -i" the terminal will display each individual inode numbers associate to each file. By using the command "find -ium(inode number)", you will be able to find that specific file with that inode number. This became a way to effectively find a certain files within the system, especially in Unix, where there are large amounts of files within the system generated by users.



```
wz345@tux2:~/public_html$ df
Filesystem            1K-blocks       Used   Available Use% Mounted on
udev                   30886608          0    30886608   0% /dev
tmpfs                   6183696       6448     6177248   1% /run
/dev/mapper/vg-root    80218632   40269228    35831492  53% /
tmpfs                  30918464     101364    30817100   1% /dev/shm
tmpfs                      5120          0        5120   0% /run/lock
tmpfs                  30918464          0    30918464   0% /sys/fs/cgroup
/dev/mapper/vg-boot      943128     219452      658552  25% /boot
/dev/mapper/vg-temp    19092136     730180    17369088   5% /tmp
ceph-fuse           70769418240 7957454848 62811963392  12% /home2
ceph-fuse           70769418240 7957454848 62811963392  12% /home
ceph-fuse           70769418240 7957454848 62811963392  12% /site
tmpfs                   6183692          8     6183684   1% /run/user/120
tmpfs                   6183692         12     6183680   1% /run/user/11325
tmpfs                   6183692          0     6183692   0% /run/user/10465
tmpfs                   6183692          0     6183692   0% /run/user/13834
tmpfs                   6183692         36     6183656   1% /run/user/15489
tmpfs                   6183692         16     6183676   1% /run/user/15724
tmpfs                   6183692         20     6183672   1% /run/user/15702
```

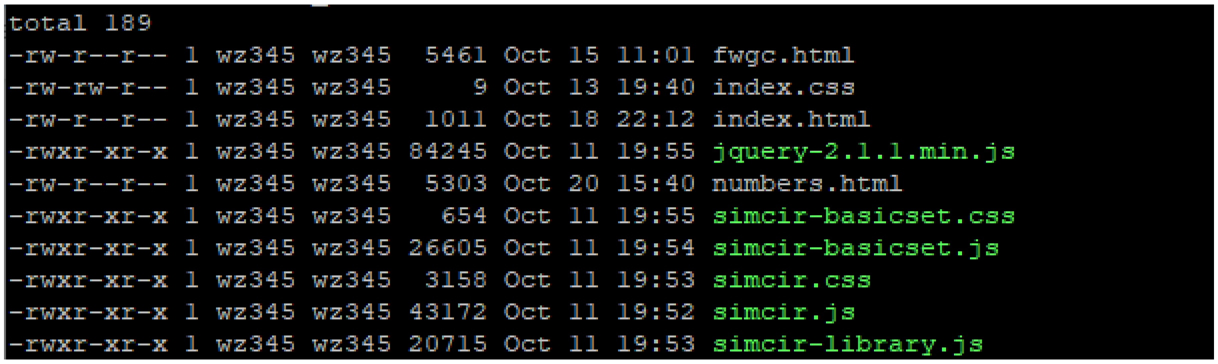
(Picture shows the inode usage after the command "df")



```
wz345@tux2:~/public_html$ ls
fwgc.html   jquery-2.1.1.min.js  simcir-basicset.js  simcir-library.js
index.css   numbers.html         simcir.css
index.html  simcir-basicset.css  simcir.js
wz345@tux2:~/public_html$ ls -i
1099619637883 fwgc.html           1099619705770 simcir-basicset.css
1099619788718 index.css           1099619705767 simcir-basicset.js
1099618719490 index.html          1099619705759 simcir.css
1099619705772 jquery-2.1.1.min.js 1099619705754 simcir.js
1099620244865 numbers.html        1099619705766 simcir-library.js
wz345@tux2:~/public_html$ find -inum 1099619705754
./simcir.js
```

(Picture shows the file within the directory, the inode number of each files after the command "ls -i" and using the "find" command to find the simcir.js file)

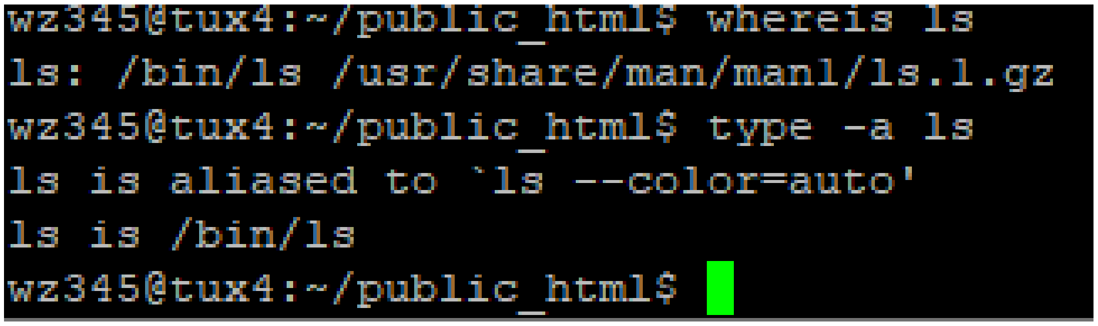
As a system built around files, information about the files in the Unix system is important too, and Unix allows us to check all the necessary information about each file by using the command "ls -l". In the information that the shell displayed after the command, the first column shows the file permission in which the first digit represents the specialness of the file, the next 3 digits shows the owner's permission, the next 3 digits after shows the permission of the group and the last 3 digits shows the permission of other users in the system. Each digit will have either a "r, w, x or -", and each digits are corresponding to a specific permission, where "r" represents the permission to read, "w" represents the permission to write, "x" represents the permission to execute, and "-" represents no permission on the corresponding digits. After that, the second column shows the amounts of link that can be used to access the file, the third and fourth column shows the owner and group of the file, the fifth column shows the size of the file, the sixth and seventh column shows the last modified date of the file and the last column shows the name of the file.



```
total 189
-rw-r--r-- 1 wz345 wz345  5461 Oct 15 11:01 fwgc.html
-rw-rw-r-- 1 wz345 wz345     9 Oct 13 19:40 index.css
-rw-r--r-- 1 wz345 wz345  1011 Oct 18 22:12 index.html
-rwxr-xr-x 1 wz345 wz345 84245 Oct 11 19:55 jquery-2.1.1.min.js
-rw-r--r-- 1 wz345 wz345  5303 Oct 20 15:40 numbers.html
-rwxr-xr-x 1 wz345 wz345   654 Oct 11 19:55 simcir-basicset.css
-rwxr-xr-x 1 wz345 wz345 26605 Oct 11 19:54 simcir-basicset.js
-rwxr-xr-x 1 wz345 wz345  3158 Oct 11 19:53 simcir.css
-rwxr-xr-x 1 wz345 wz345 43172 Oct 11 19:52 simcir.js
-rwxr-xr-x 1 wz345 wz345 20715 Oct 11 19:53 simcir-library.js
```

(Picture above shows the file information after the "ls -l" command)

In Unix, everything is within the file system, even the commands that the users type in the shell. By using the command "whereis" the user will be able to find where the targeted executable command is located. This path is also used when the system tries to look for the command after the user type a command in the shell. There is also a way to get more information about the executable, using the command "ls" for example to get more information about the executable command, the command “type -a ls” can be used to check both the location of the executable and its alias. An alias is basically a shortcut to access the command, it was made so that it can access the executable without typing the full original command. But in this case, the alias for “ls” is longer than its original. Although the “whereis” and “type -a” commands are useful, but it only can be used to find executables, if it is used otherwise like finding an html file, it will just display a blank output.



```
wz345@tux4:~/public_html$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
wz345@tux4:~/public_html$ type -a ls
ls is aliased to `ls --color=auto'
ls is /bin/ls
wz345@tux4:~/public_html$
```

(Picture shows the effect of the “whereis” command and the “type -a ls” command)

Unix is an important system in the history of operating systems, many modern operating systems have their roots in Unix. It is a unique operating system that was built around a file system and the concept of time sharing. As the most important feature in Unix, the file system of Unix is special on its own. It has the ilist, where it contains inodes that is associated with every file in the disk, so that the system knows where these files are. There is a shell where the users need to use commands instead of normal interactions to communicate with the system. There are commands that can do anything a user wants like showing the details of the file or the location of the file. But it doesn’t just end there, since the system is made up of files, the processes of the system can also be monitored and traced back to where it is located, even the executable commands that are used by the user can be located by using commands itself. At the end, the file system of Unix is truly special, although it might not be as visually pleasing because it doesn’t have icons or visual cues to identify the files, but the functionality and freedom of the file system will deliver a completely different experience than the other operating systems.

Cited Page

“It Is All about the Inode.” *IBM Developer*, https://developer.ibm.com/articles/au-speakingunix14/.

“The Evolution of the Unix Time-Sharing System*.” *Early Unix History and Evolution*, https://www.bell-labs.com/usr/dmr/www/hist.html.

Ashok, Arunlal. "What Is Inode Number in Linux?" *Crybit.com*, 19 Nov. 2018, https://www.crybit.com/inode-number-linux/.

"Hierarchical File Structure." *UNIX File System*, http://www.cis.rit.edu/class/simg211/unixintro/Filesystem.html.

"Unix / Linux - File Permission / Access Modes." *Tutorialspoint*, https://www.tutorialspoint.com/unix/unix-file-permission.html.

The Unix File System, http://homepages.uc.edu/~thomam/Intro_Unix_Text/File_System.html.

Byrd, Jonathan. *The Unix File System*, https://fsl.fmrib.ox.ac.uk/fslcourse/unix_intro/fstour.html.

Origins and History of Unix, 1969-1995, https://homepage.cs.uri.edu/~thenry/resources/unix_art/ch02s01.html.

Gite, Vivek, et al. "Linux / UNIX: Determine Where a Binary Command Is Stored / Located on File System." *NixCraft*, 29 Aug. 2007, https://www.cyberciti.biz/tips/find-linux-unix-command-location.html.