CENTRAL CALCUTTA POLYTECHNIC

21, Convent Road, Philips, Sealdah, Kolkata, West Bengal 700014

DEPT. : COMPUTER SCIENCE AND TECHNOLOGY

- NAME : SUMAN MONDAL
- ROLL : DCCPCSTS5
- NUMBER : 10005537
- REG NUMBER : D192005242
- SUBJECT : OPERATING SYSTEM
- SESSION : 2021 - 2022
- EMAIL : SUMAN.MONDAL@OUTLOOK.IN

1 Operating System Assignment

a. Describe the Architecture of UINX OS

The main concept that unites all the versions of Unix is the following four basics –

- **Kernel** – The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.
- **Shell** – The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.
- **Commands and Utilities** – There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.
- **Files and Directories** – All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the filesystem.

b. Features of UNIX OS

Unix is an operating system, so it has all the features that the OS must-have. UNIX also looks at a few things in a different way than other OS. Features of UNIX are listed below :

- **Multiuser System** : In UNIX, resources are actually shared between all the users, so-called a multi-user system. For doing so, computer give a time slice (breaking unit of time into several segments) to each user. So, at any instant of time, only one user is served but the switching is so fast that it gives an illusion that all the users are served simultaneously.
- **The Building-Block Approach** : The Unix developers thought about keeping small commands for every kind of work. So Unix has so many commands, each of which performs one simple job only. You can use 2 commands by using pipes (‘|’).
- **The UNIX Toolkit** : Unix has a kernel but the kernel alone can't do much that could help the user. So, we need to use the host of applications that usually come along with the UNIX systems. The applications are quite diversified.
- **Pattern Matching** : Unix provides very sophisticated pattern matching features. The meta-char ‘*’ is a special character used by the system to match a number of file names. There are several other meta-char in UNIX.
- **Programming Facility** : Unix provides shell which is also a programming language designed for programmers, not for casual end-users. It has all the control structures, loops, and variables required for programming purposes. These features are used to design the shell scripts (programs that can invoke the UNIX commands).

  Many functions of the system can be controlled and managed by these shell scripts.

c. Difference between UNIX, WINDOWS and DOS

- UNIX

(a) Supports both CLI/GUI
(b) Supports Multitasking
(c) Unix can be installed on a wide variety of computer hardware, mobile phones and video game consoles to mainframes and supercomputers
(d) Very reliable and is known for being run for months or even years without needing to be rebooted
(e) Difficult to manage

- WINDOWS

(a) Supports GUI
(b) Supports Multitasking
(c) Windows can be installed on PC desktops, laptops, servers, and some mobile phones
(d) Windows is not even reliable, with it crashing/needing to be restarted all the time
(e) Easy to manage

- DOS

(a) Supports a Text-Based/CLI
(b) DOS is unable to run multiple processes at the same time
(c) DOS can be installed on standalone PC desktops or laptops
(d) DOS is not very reliable because easily if a prgram doesn't open properly or a wrong command is issued.
(e) DOS is not very easy to manage

d. Explain the booting process of an operating system

When our computer is switched on, it can be started by hardware such as a button press, or by software command, a computer's central processing unit (CPU) has no software in its main memory, there is some process which must load software into main memory before it can be executed. Below are the six steps to describe the boot process in the operating system, such as:

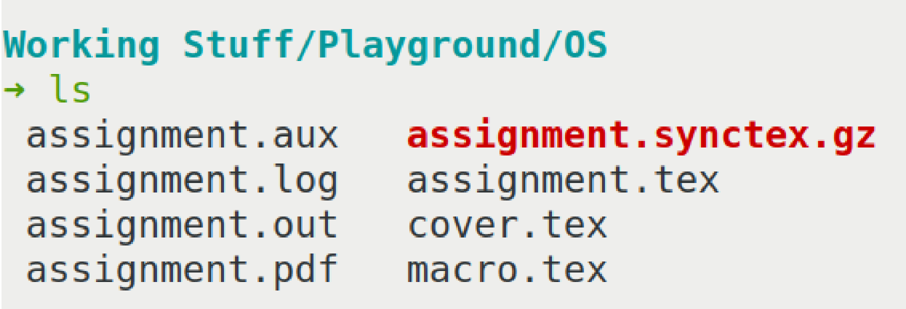
- **Step 1** : Once the computer system is turned on, BIOS (Basic Input /Output System) performs a series of activities or functionality tests on programs stored in ROM, called on POST (Power-on Self Test) that checks to see whether peripherals in the system are in perfect order or not.
- **Step 2** : After the BIOS is done with pre-boot activities or functionality test, it read bootable sequence from CMOS (Common Metal Oxide Semiconductor) and looks for master boot record in the first physical sector of the bootable disk as per boot device sequence specified in CMOS. For example, if the boot device sequence is:
- **Step 3** : The master boot record will search first in a floppy disk drive. If not found, then the hard disk drive will search for the master boot record. But if the master boot record is not even present on the hard disk, then the CDROM drive will search. If the system cannot read the master boot record from any of these sources, ROM displays "No Boot device found" and halted the system. On finding the master boot record from a particular bootable disk drive, the operating system loader, also called Bootstrap loader, is loaded from the boot sector of that bootable drive into memory. A bootstrap loader is a special program that is present in the boot sector of a bootable drive.

- **Step 4** : The bootstrap loader first loads the IO.SYS file. After this, MSDOS.SYS file is loaded, which is the core file of the DOS operating system.
- **Step 5** : After this, MSDOS.SYS file searches to find Command Interpreter in CONFIG.SYS file, and when it finds, it loads into memory. If no Command Interpreter is specified in the CONFIG.SYS file, the COMMAND.COM file is loaded as the default Command Interpreter of the DOS operating system.
- **Step 6** : The last file is to be loaded and executed is the AUTOEXEC.BAT file that contains a sequence of DOS commands. After this, the prompt is displayed. We can see the drive letter of bootable drive displayed on the computer system, which indicates that the operating system has been successfully on the system from that drive.

e. Use different command and also write the output in Unix : ls, rm, mv, cp, cat, comm, chmod, mkdir, rmdir, cd, pwd, cmp, sort, grep, who am i, date, time, cal, clear, man, join, split, head, tail, diff

1.1 ls - show content of the directory

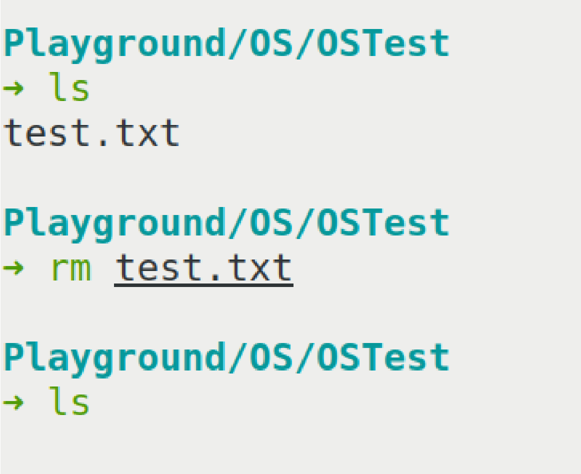
Output :



```
Working Stuff/Playground/OS
➜ ls
 assignment.aux   assignment.synctex.gz
 assignment.log   assignment.tex
 assignment.out   cover.tex
 assignment.pdf   macro.tex
```

1.2 rm - to remove file

Output :



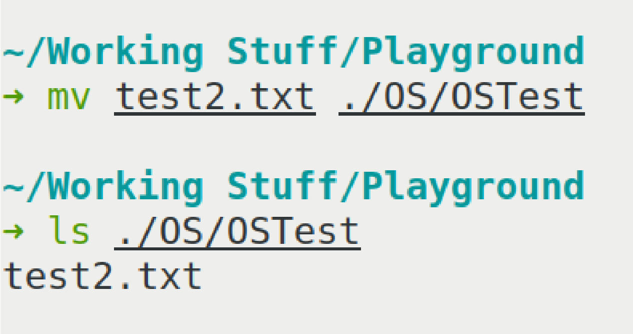
```
Playground/OS/OSTest
➜ ls
test.txt

Playground/OS/OSTest
➜ rm test.txt

Playground/OS/OSTest
➜ ls
```

1.3 mv - move content to another location

Output :

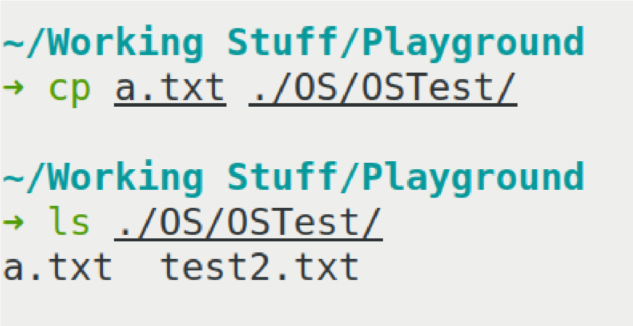


```
~/Working Stuff/Playground
→ mv test2.txt ./OS/OSTest

~/Working Stuff/Playground
→ ls ./OS/OSTest
test2.txt
```

1.4 cp - copy content(s) to another location

Output :

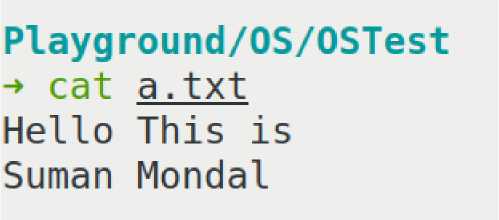


```
~/Working Stuff/Playground
→ cp a.txt ./OS/OSTest/

~/Working Stuff/Playground
→ ls ./OS/OSTest/
a.txt  test2.txt
```

1.5 cat - display the content of the file

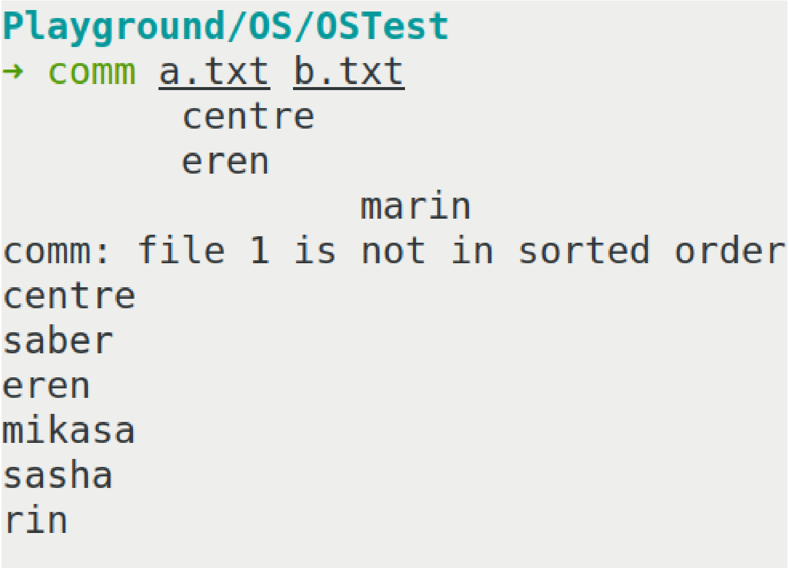
Output :



```
Playground/OS/OSTest
→ cat a.txt
Hello This is
Suman Mondal
```

1.6 comm - compares two sorted files line by line

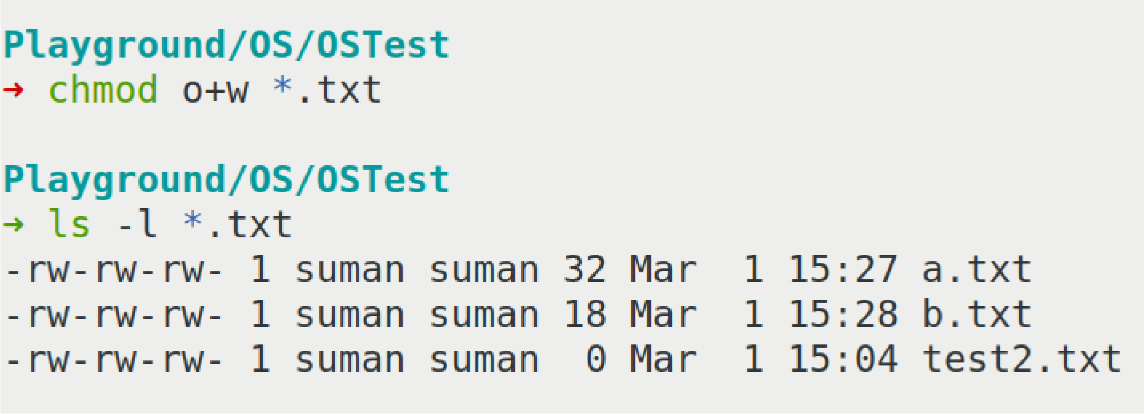
Output :



```
Playground/OS/OSTest
→ comm a.txt b.txt
        centre
        eren
                marin
comm: file 1 is not in sorted order
centre
saber
eren
mikasa
sasha
rin
```

1.7 chmod - to change the access permissions of files and directories

Output :

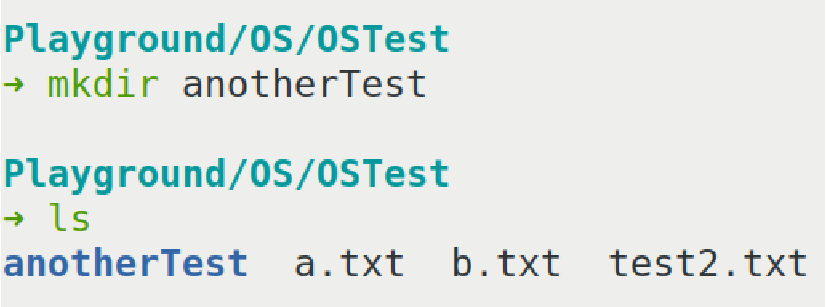


```
Playground/OS/OSTest
→ chmod o+w *.txt

Playground/OS/OSTest
→ ls -l *.txt
-rw-rw-rw- 1 suman suman 32 Mar  1 15:27 a.txt
-rw-rw-rw- 1 suman suman 18 Mar  1 15:28 b.txt
-rw-rw-rw- 1 suman suman  0 Mar  1 15:04 test2.txt
```

1.8 mkdir - make directory

Output :

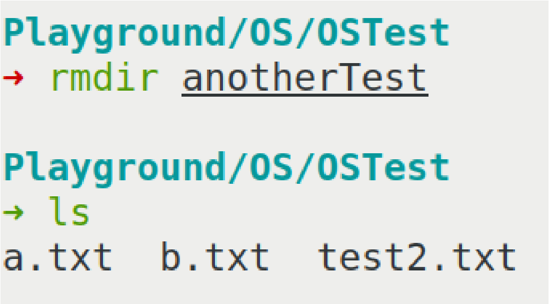


```
Playground/OS/OSTest
→ mkdir anotherTest

Playground/OS/OSTest
→ ls
anotherTest  a.txt  b.txt  test2.txt
```

1.9 rmdir - remove directory

Output :

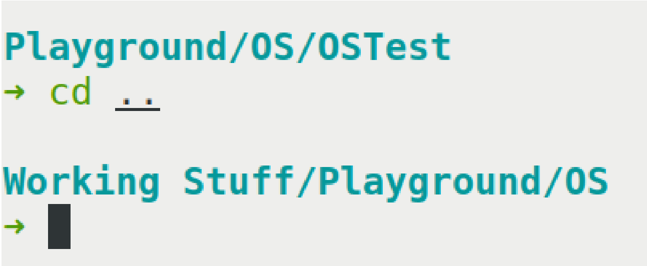


```
Playground/OS/OSTest
➜ rmdir anotherTest

Playground/OS/OSTest
➜ ls
a.txt  b.txt  test2.txt
```

1.10 cd - change directory path

Output :

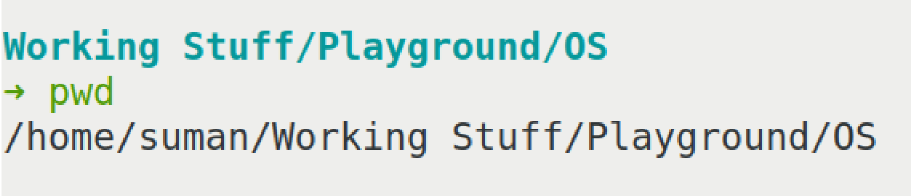


```
Playground/OS/OSTest
➜ cd ..

Working Stuff/Playground/OS
➜ █
```

1.11 pwd - print the working directory

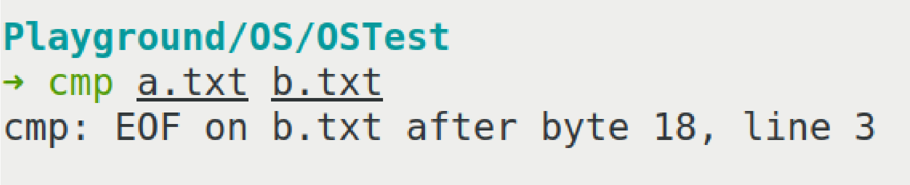
Output :



```
Working Stuff/Playground/OS
➜ pwd
/home/suman/Working Stuff/Playground/OS
```

1.12 cmp - compare to files

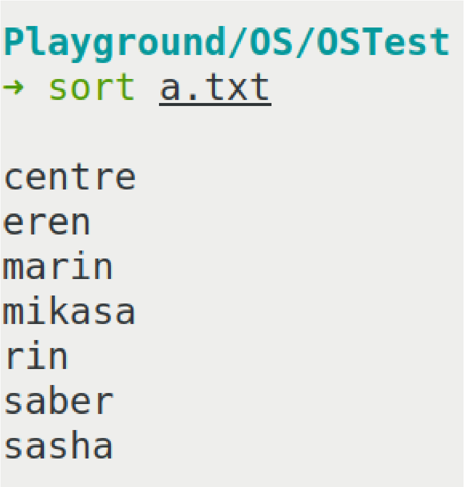
Output :



```
Playground/OS/OSTest
➜ cmp a.txt b.txt
cmp: EOF on b.txt after byte 18, line 3
```

1.13 sort - sorts the file content in an alphabetical order

Output :

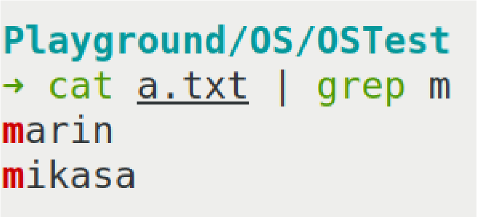


```
Playground/OS/OSTest
➜ sort a.txt

centre
eren
marin
mikasa
rin
saber
sasha
```

1.14 grep - filters the content of a file which makes our search easy

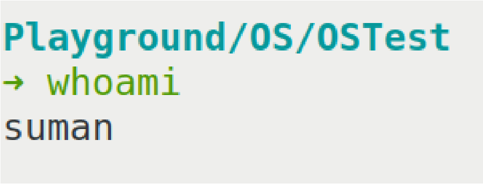
Output :



```
Playground/OS/OSTest
➜ cat a.txt | grep m
marin
mikasa
```

1.15 whoami - shows current user when this command will be invoked

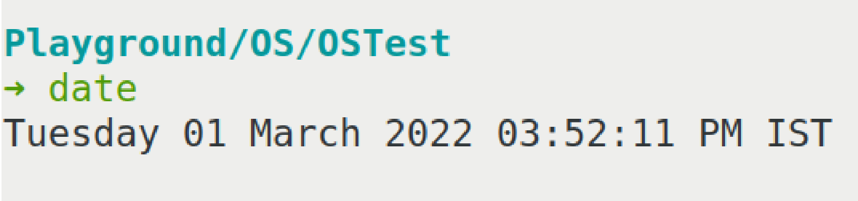
Output :



```
Playground/OS/OSTest
➜ whoami
suman
```

1.16 date - shows current date

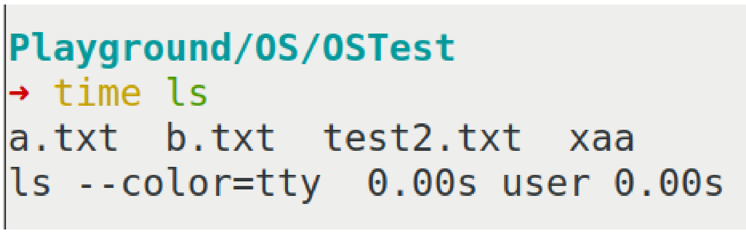
Output :



```
Playground/OS/OSTest
➜ date
Tuesday 01 March 2022 03:52:11 PM IST
```

1.17 time - displays how long it takes to execute a command

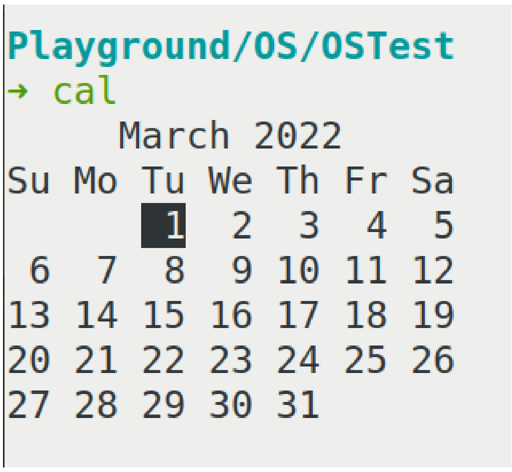
Output :



```
Playground/OS/OSTest
➜ time ls
a.txt  b.txt  test2.txt  xaa
ls --color=tty  0.00s user 0.00s
```

1.18 cal - shows current month's calender with current day highlighted

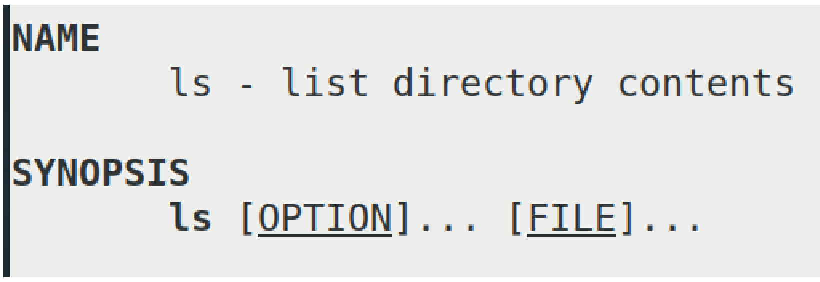
Output :



```
Playground/OS/OSTest
➜ cal
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

1.19 man - shows the manual page of command

Output :

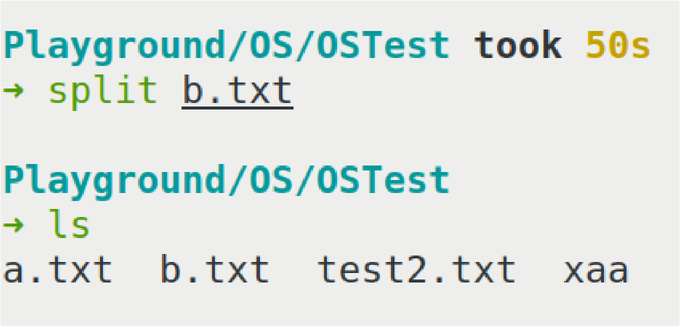


```
NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...
```

1.20 split - split larger files into smaller files

Output :

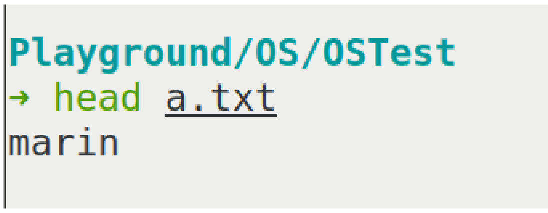


```
Playground/OS/OSTest took 50s
→ split b.txt

Playground/OS/OSTest
→ ls
a.txt  b.txt  test2.txt  xaa
```

1.21 head - displays the starting content of a file. By default, it displays starting 10 lines of any file

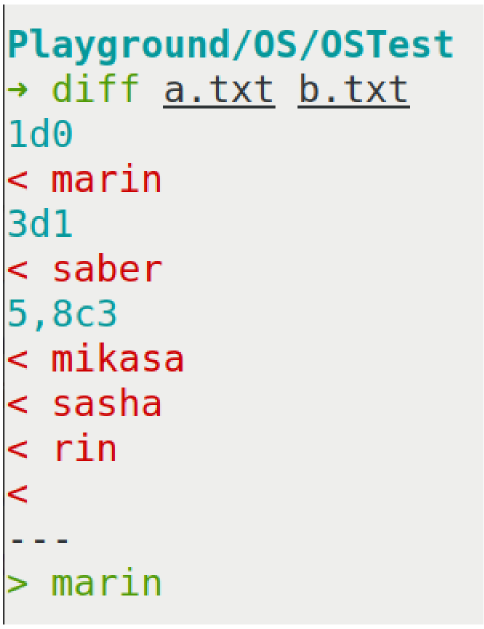
Output :



```
Playground/OS/OSTest
→ head a.txt
marin
```

1.22 diff - display the differences in the files by comparing the files line by line

Output :



```
Playground/OS/OSTest
→ diff a.txt b.txt
1d0
< marin
3d1
< saber
5,8c3
< mikasa
< sasha
< rin
<
---
> marin
```