Tutorial 2: Unix Command Line (II)

CS 108

Spring, 2023-24

TA: Guramrit Singh

Topics

- Redirection and pipes
- Process Management
- Access Control
- Different Users
- Regular Expressions
- Some other commands
- Exercises

Redirection and pipes

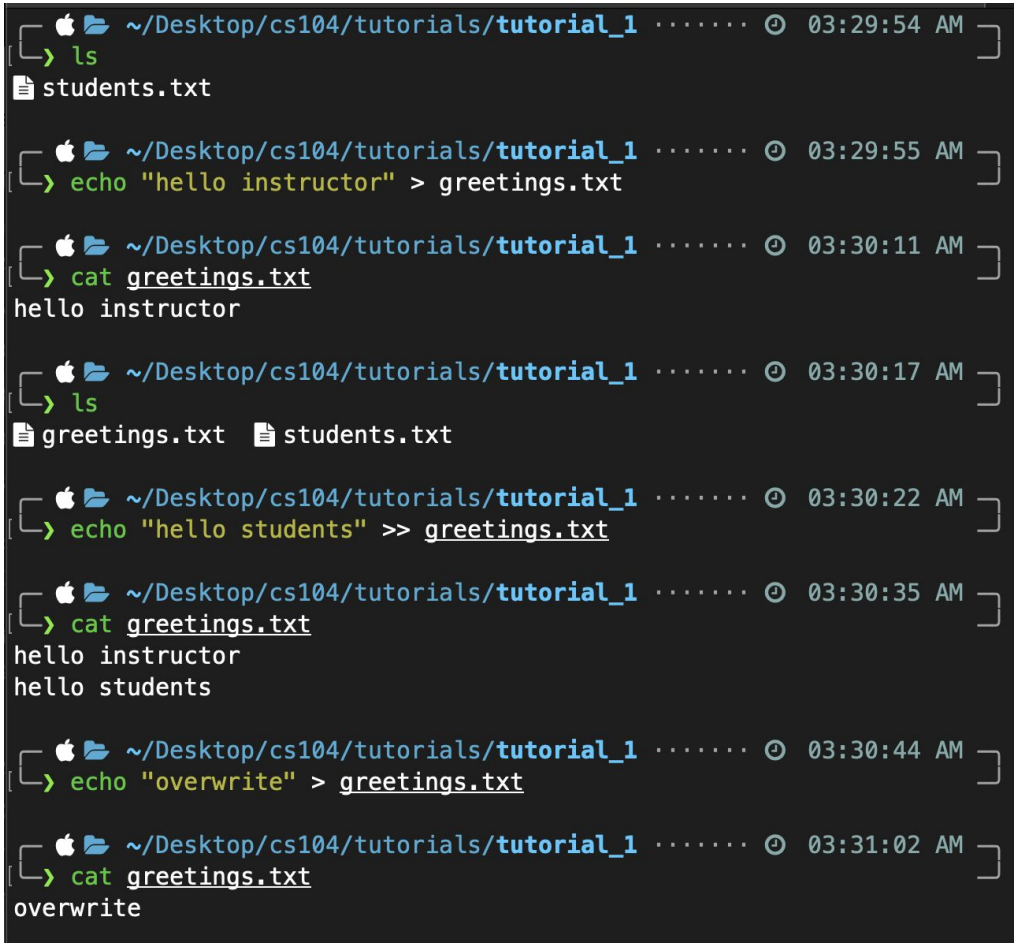
- >
- <
- <<
- |

Redirection

- Frequently, there's a desire to avoid manual input of command arguments or to save output to a file.
- Redirection operators offer a convenient way to accomplish these tasks.
- Types of redirection operators:
  - `>` : Redirects standard output to a file, overwriting the file if it already exists.
  - `>>` : Redirects standard output to a file, appending the output to the end of the file if it already exists.
  - `<` : Redirects standard input to come from a file.

>, >>

- We saw echo command before. Now, we would like to redirect the output of echo command to a file rather than the terminal.
- In the example,
  - We first redirect “hello instructor” to greetings.txt
  - Then we append “hello students” to greetings.txt
  - Then we overwrite the file using > operator



Pipes

```
~/Desktop/cs104/tutorials/tutorial_1 ····· 03:52:07 AM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME

~/Desktop/cs104/tutorials/tutorial_1 ····· 03:52:09 AM
❯ cat students.txt | head -3
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE

~/Desktop/cs104/tutorials/tutorial_1 ····· 03:52:19 AM
❯ cat students.txt | head -3 | tail -1
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
```

- Often, there's a preference to bypass manually specifying command arguments or saving output to a file, opting instead to seamlessly pass the output of one command as the input to another using a pipe.
- | : Redirects the output of one command as the input to another command (pipe).
- In the example, we want to find the third line of the students.txt file.
  - We first use cat on students.txt file.
  - Then, we pass the above output to head, which extracts first three lines.
  - Then, we pass the above output to tail, which extracts the last line, and thus giving us the desired output.

Process Management

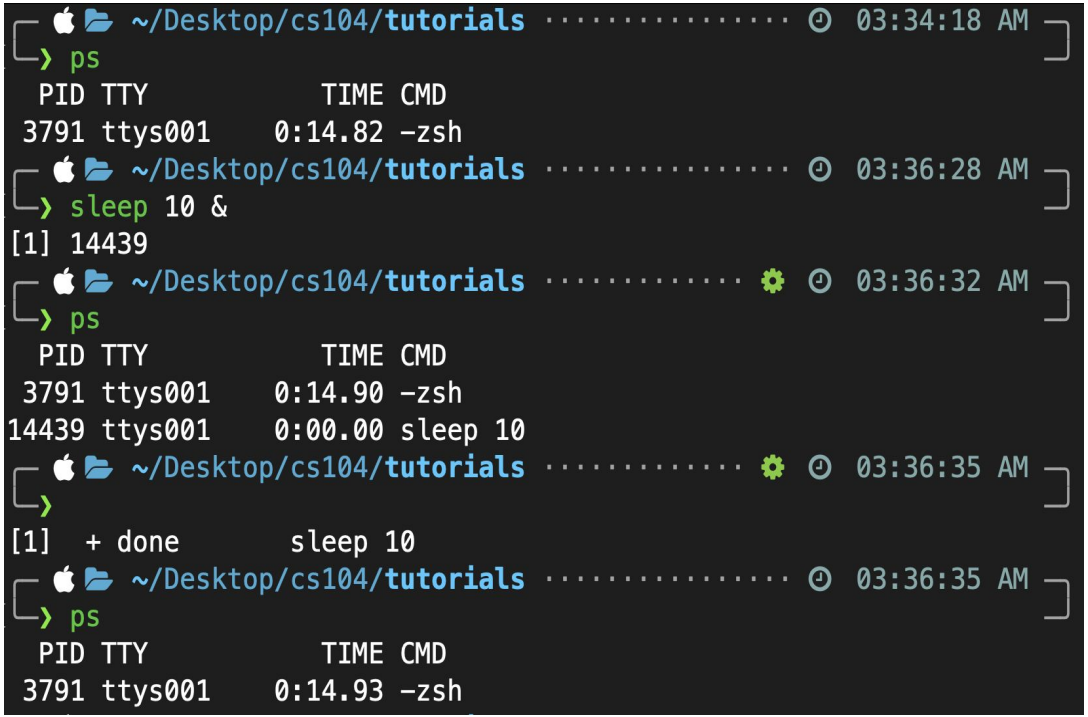
- ps
- kill

Processes

- A program in execution is referred to as process.
- It consists of several components, including data retrieved from files, user input, program instructions, etc.
- Same program can be executed any number of times, each execution instance becomes a new process.
- Each process has a unique id referred to as pid.
- init is the first process that is created by OS during boot up and usually has pid 1.
- Every other process is created by another process referred to as parent process and it's pid referred to as ppid.
- Interested in processes?? More on this in your OS course!!!
- Next we discuss two frequently used tools ps and kill dealing with processes

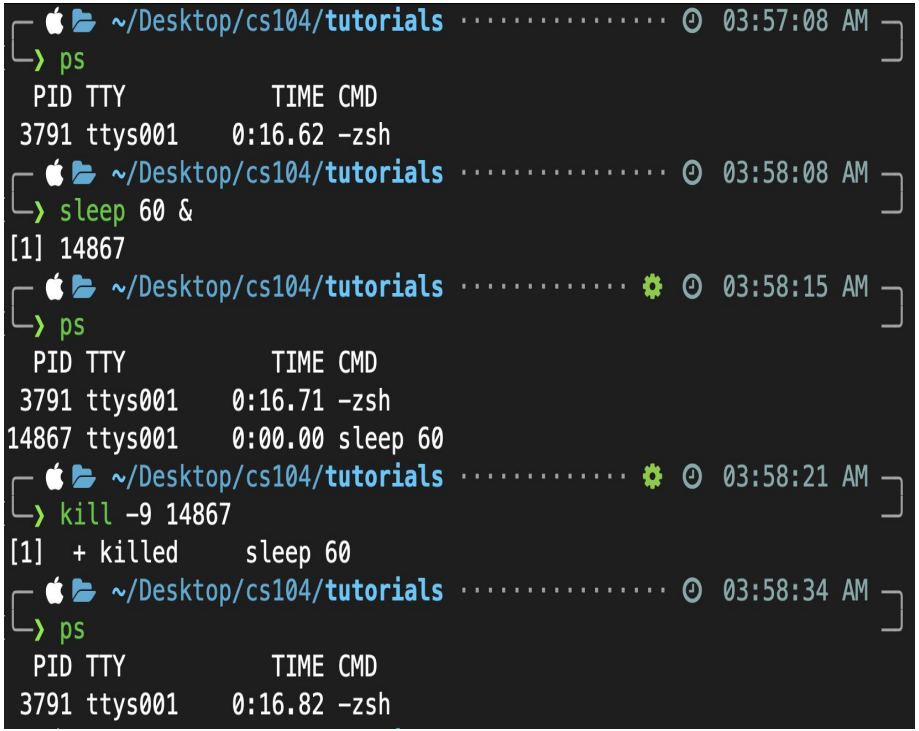
ps

- ps : process status, displays a header line, followed by lines containing information about all of your processes that have controlling terminals.
- Some of the useful options are:
  - -a : Display info about processes of other users' as well as yours.
  - -u : Display the processes belonging to the specified usernames.
  - Checkout : -f, -x, -j, -m



Example: We first use ps and see that only shell process is running on the current terminal, then we execute sleep for 10 seconds in background and see the ps output before and after sleep process terminates.

kill



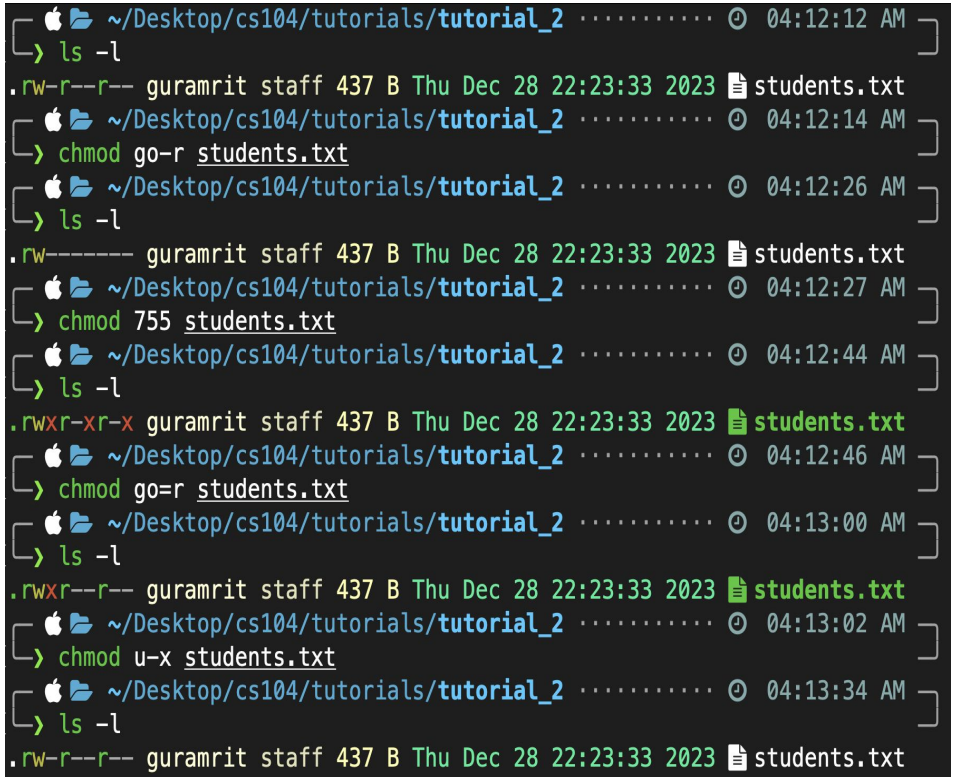
Example: We execute sleep for 60 seconds in background and see the ps output before and after kill command terminates the sleep process.

- The kill utility sends a signal to the processes specified by the pid operands. (default is TERM)
- Only the super-user may send signals to other users' processes.
- Some of the useful options are:
  - -s : signal name specifying the signal to be sent.
  - -signal_number : A non-negative decimal integer, specifying the signal to be sent.
  - Some signal numbers and symbols
    - 2 - INT (interrupt)
    - 9 - KILL (non-ignorable kill)
    - 15 - TERM (software termination)
  - Checkout other signal numbers and signal symbols mapping. (Look at -l option)

Access Control

- chmod

chmod



- The chmod utility modifies the file mode bits of the listed files as specified by the mode operand.
- Symbolic Mode:
  - u - The file owner.
  - g - The users who are members of the group.
  - o - All other users.
  - a - All users, identical to ugo
  - - Removes the specified permissions.
  - + Adds specified permissions.
  - = Changes the current permissions to the specified permissions
    - If no permissions are specified after the = symbol, all permissions from the specified user class are removed

Example: 1. By using go-r, we removed read access from group and others. 2.By using 755 (111 101 101), we gave rwx to user, and rx to group and others. By using go=r, we changed access of group, others to read only. 4. Finally, by using u-x we removed execute permission from user.

Different Users

- su
- sudo

su

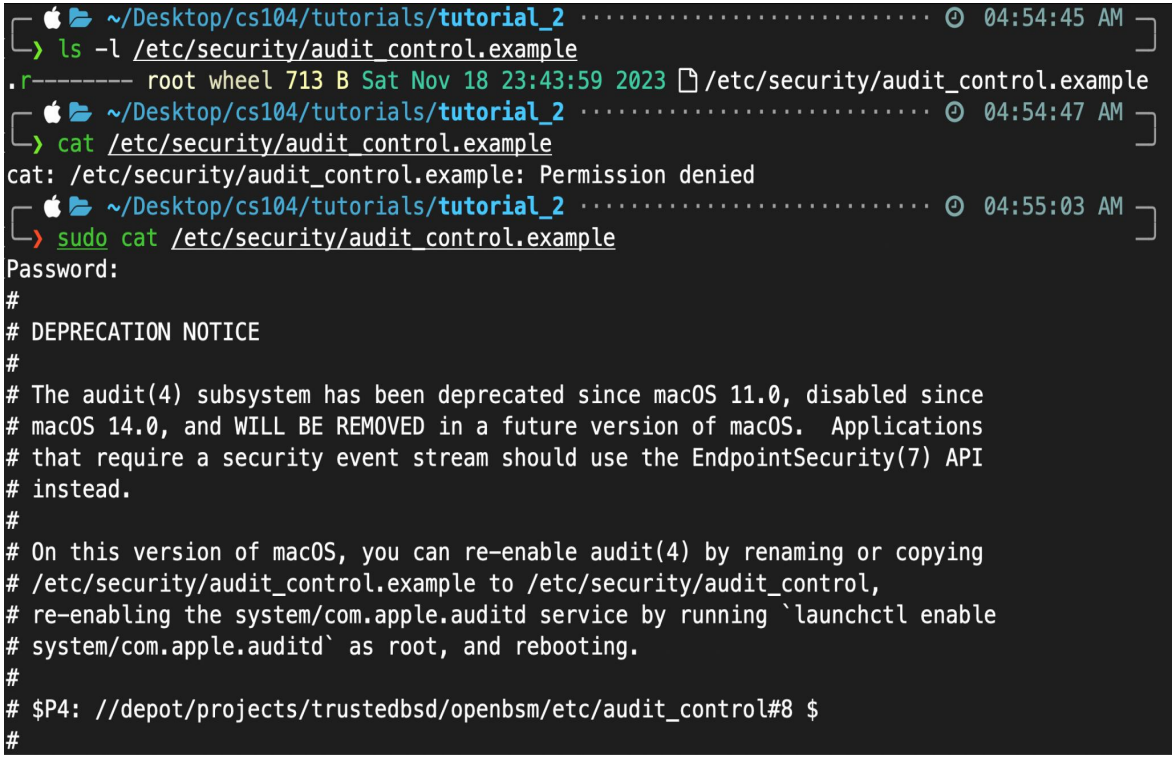
The su utility requests appropriate user credentials and switches to that user ID (the default user is the superuser). A shell is then executed.

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 04:42:30 AM
❯ ls -l
.rw-r--r-- guramrit staff 437 B Thu Dec 28 22:23:33 2023 students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 04:49:11 AM
❯ su cs104
Password:
Restored session: Fri Dec 29 04:42:24 IST 2023
cs104@192 . % cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
cs104@192 . % echo 0 > students.txt
zsh: permission denied: students.txt
cs104@192 . % exit
```

Example: We saw the access controls, now lets test them. We are logged in as guramrit, we will use su to switch to cs104 account. Note that students.txt (owned by guramrit/staff) file had read only permissions for group and others and so we will be able to use cat but won't be able to write to it using echo.

sudo

The sudo allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.



Example: We will try to read a file with owner as root and no access to group and others and as you would expect, we fail to read the file. Trying the same command with sudo, lets us read the file.

Regular Expressions

- grep

Basics of regex

- Regex is a powerful tool for finding text according to a particular pattern in a variety of situations. We will use it in grep (today) and sed, awk (in some later week).
- There are three basic building blocks when working with regular expressions: *regular characters*, *metacharacters*, and *patterns*.
- Meta-characters (some of them are listed below):
  - ^ : start of a line (NOTE: Can also mean "not" if used inside [])
  - $: end of line
  - . : match any single character
  - \ : escape a special character
  - | : logical OR operation i.e. match a particular character set on either side
  - * : search for a character that occurs zero or more times as defined by the preceding character
  - + : search for a character that occurs one or more times as defined by the preceding character
  - ? : search for a character that occurs zero or one time as defined by the preceding character
  - \d : represents any single numeral, 0 through 9
  - \s : represents space

Basics of regex

- A quantifier is a syntactic structure in regular expressions that indicates the number of times a character occurs in sequence in the input text.
- Some of them are listed below:
  - {n} : the preceding character needs to occur exactly n times
  - {n,} : the preceding character needs to occur at least n times
  - {n,m} : the preceding character needs to occur between n and m times
- Groups and ranges:
  - (<def>) : a group of characters declared according to a specific definition
  - [<range>] : match any character from range of given characters in the []
    - [0-9] : match any digit from 0-9
    - [a-z] : match any lowercase letter from a-z
    - [A-Z] : match any uppercase letter from A-Z
  - [^<range>] : match any character not in the range of given characters in the []
    - [^adA-Z] : match any character which is not a, d or lies in A-Z

grep

- The grep utility searches any given input files, selecting lines that match one or more patterns.
- Some of the useful options are:
  - -i : Case insensitive match (default is case sensitive)
  - -e : Specify a pattern used during the search of the input (you can use -e any number of times)
  - -o : Prints only the matching part of the lines.
  - Checkout options: -n, -v, -m

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:17:11 PM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:17:14 PM
❯ tail -n +2 students.txt| grep -e ',..$'
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:17:17 PM
❯ tail -n +2 students.txt| grep -e ',[kgs].*[^a]@'
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:17:25 PM
❯ grep -e 'guramrit' students.txt
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:17:57 PM
❯ grep -e 'guramrit' -i students.txt
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:18:03 PM
❯ tail -n +2 students.txt| grep -e '^21'
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
```

Example: 1. All students from department having shorthand of 2 characters. 2. Students with e-mail id starting with k/g/s and ending not with a. 3. Case insensitive search. 4. Students of 2021 batch

Some other commands

- cut
- wc
- diff
- sort
- tar
- zip/unzip

cut

- cut out selected portions of each line of a file
- Some of the useful options are:
  - -d : Used to specify field delimiter character (default is tab)
  - -f : Used to specify the fields desired in the output and separated in the input by field delimiter character.

```
~/Desktop/cs104/tutorials/tutorial_2 · 01:28:57 AM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 · 01:29:55 AM
❯ cut -d ',' -f 2,6 students.txt
Name,Department
Guramrit Singh,CSE
Akshay Kumar,EE
Kiara Advani,EP
Kavya Gupta,CSE
Saksham Rathi,CSE
Rashmika Mandanna,EE
Harmanpreet Kaur,ME
```

Example: From students.txt file, we extracted 2nd and 6th columns corresponding to name and department.

wc

- The wc utility displays the number of lines, words, and bytes contained in each input file
- Some of the useful options are:
  - -l : Used to get the number of lines in each input file
  - -w : Used to get the number of words in each input file
  - Checkout options : -L, -c

```
~/Desktop/cs104/tutorials/tutorial_2 · 01:43:47 AM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 · 01:44:00 AM
❯ wc students.txt
       8      15     437 students.txt
~/Desktop/cs104/tutorials/tutorial_2 · 01:44:04 AM
❯ wc -l students.txt
       8 students.txt
```

Example: By default wc outputs #lines, #words, #characters present in the input files.

diff

- The diff utility compares the contents of file1 and file2 and writes to the standard output the list of changes necessary to convert one file into the other.
- No output is produced if the files are identical.
- Checkout options : -B, -w

```
~/Desktop/cs104/tutorials/tutorial_2 · 02:14:11 AM
❯ cp test.txt test_repl.txt
~/Desktop/cs104/tutorials/tutorial_2 · 02:14:21 AM
❯ diff test.txt test_repl.txt
~/Desktop/cs104/tutorials/tutorial_2 · 02:14:36 AM
❯ vim test_repl.txt
~/De/c/tutorials/tutorial_2 ··· 12s 02:15:04 AM
❯ cat test.txt
first
second
third
~/Desktop/cs104/tutorials/tutorial_2 · 02:15:11 AM
❯ cat test_repl.txt
fast
third
~/Desktop/cs104/tutorials/tutorial_2 · 02:15:17 AM
❯ diff test.txt test_repl.txt
1,2c1
< first
< second
---
> fast
```

Example: First we replicated test.txt to test_repl.txt, observe that diff produced no output, then we deleted second line and changed first line of the replicated file , see the diff output.

sort

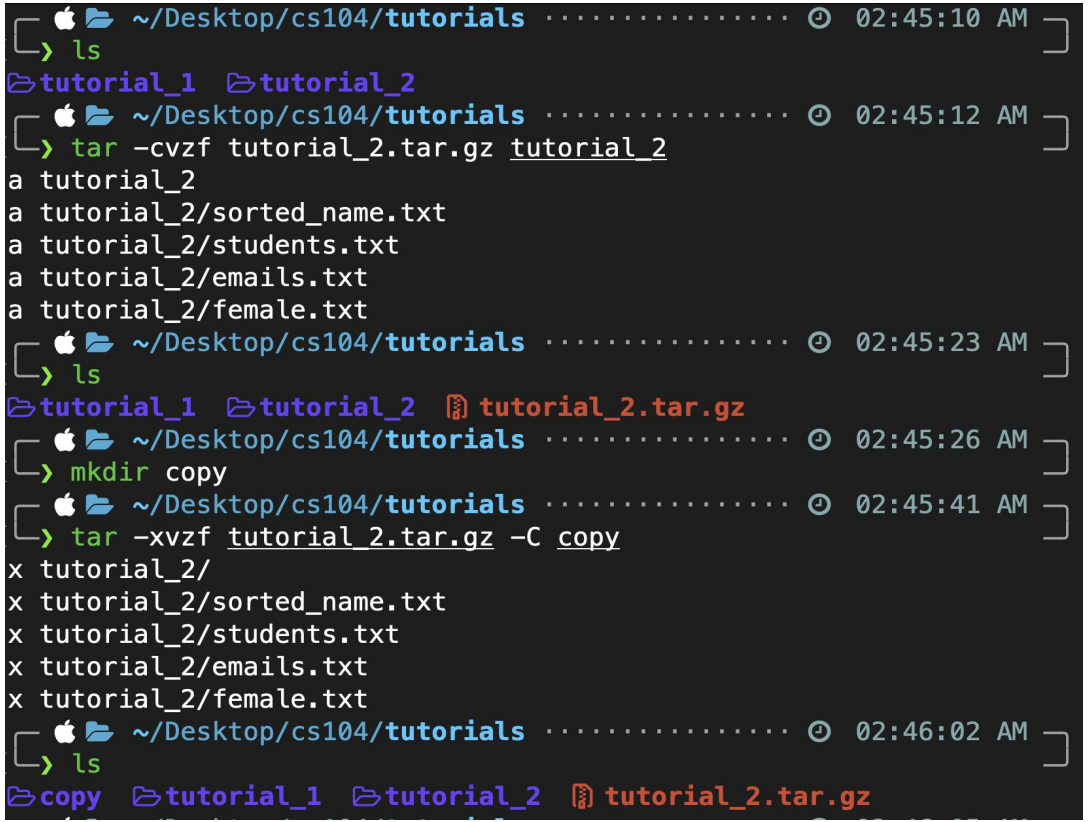
- The sort utility sorts text and binary files by lines. A line is a record separated from the subsequent record by a newline (default).
- Some of the useful options are:
  - -t : Used to specify field separator character.
  - -r : Used to sort in reverse order.
  - -k : Used to specify the field(s) that will be used as sort key(s).
  - Checkout options : -c

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:30:08 AM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:30:11 AM
❯ tail -n +2 students.txt | sort -t ',' -k 1
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
```

Example: Sort the students (leaving the header) in students.txt according to roll numbers.

tar

- tar creates and manipulates streaming archive files.
- To tar a folder, we will use the following command
  `tar -cvzf <output file> <folder>`
- To untar, we will use the following command
  `tar -xvzf <tar file>`
- Checkout the above options if you are curious, namely
  -c, -v, -x, -z, -f



Example: We first tar tutorial_2, then we untar it into copy folder using -C option, (by default pwd is used for untarring)

zip/unzip

- zip is a compression and file packaging utility for Unix
- To zip a folder recursively, we will use the following command
  `zip -r <zip file> <folder>`
- unzip - list, test and extract compressed files in a zip archive
- To unzip, we will use the following command
  `unzip <zip file>`

```
~/Desktop/cs104/tutorials ····· 03:01:25 AM
❯ ls
tutorial_1  tutorial_2
~/Desktop/cs104/tutorials ····· 03:01:28 AM
❯ zip -r tutorial_2.zip tutorial_2/
  adding: tutorial_2/ (stored 0%)
  adding: tutorial_2/sorted_name.txt (deflated 44%)
  adding: tutorial_2/students.txt (deflated 44%)
  adding: tutorial_2/emails.txt (deflated 52%)
  adding: tutorial_2/female.txt (stored 0%)
~/Desktop/cs104/tutorials ····· 03:01:32 AM
❯ ls
tutorial_1  tutorial_2  tutorial_2.zip
~/Desktop/cs104/tutorials ····· 03:01:36 AM
❯ unzip tutorial_2.zip -d copy
Archive:  tutorial_2.zip
   creating: copy/tutorial_2/
  inflating: copy/tutorial_2/sorted_name.txt
  inflating: copy/tutorial_2/students.txt
  inflating: copy/tutorial_2/emails.txt
 extracting: copy/tutorial_2/female.txt
~/Desktop/cs104/tutorials ····· 03:01:44 AM
❯ ls
copy  tutorial_1  tutorial_2  tutorial_2.zip
```

Example: We first zip tutorial_2, then we untar it into copy folder using -d option, (by default pwd is used for unzipping)

Exercises

Exercise 1

Create a file students.txt with the following content using commands learnt so far.

```
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
```

Solution 1

We use echo with redirection operator to accomplish the task.

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:21:23 PM
❯ echo "ID,Name,E-mail,Gender,Year,Department" > students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:21:34 PM
❯ echo "210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE" >> students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:21:37 PM
❯ echo "200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE" >> students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:21:54 PM
❯ echo "210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP" >> students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:22:26 PM
❯ echo "22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE" >>students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:22:42 PM
❯ echo "22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE" >> students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:23:04 PM
❯ echo "22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE" >> students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:23:19 PM
❯ echo "22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME" >> students.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:23:33 PM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
```

Exercise 2

Given the file students.txt, create a file named email.txt storing email ids of all the students

Solution 2

To accomplish the task, we do the following:

1. Use tail with *(-n +2)*, so as to get rid of first line.
2. Use cut to break the line into tokens with delimiter set to ‘,’ and take the 3rd column, and redirect it to emails.txt file.

Exercise 3

Create a file named sorted_name.txt from the contents of students.txt, arranging student information with the header intact. Ensure that the names of the students are organized in lexicographical order.

Solution 3

To accomplish the task, we do the following:

1. Firstly, using head, we copy the header to sorted_name.txt
2. Then using tail with *(-n +2)*, we get rid of header, then sort these lines based on 2nd column delimited by ‘,’ and finally redirecting output to sorted_name.txt

```
~/Desktop/cs104/tutorials/tutorial_2 ···· 10:35:10 PM
❯ head -n 1 students.txt > sorted_name.txt
~/Desktop/cs104/tutorials/tutorial_2 ···· 10:35:40 PM
❯ tail -n +2 students.txt | sort -t ',' -k 2 >> sorted_name.txt
~/Desktop/cs104/tutorials/tutorial_2 ···· 10:35:53 PM
❯ cat sorted_name.txt
ID,Name,E-mail,Gender,Year,Department
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
```

Exercise 4

Given the file students.txt, create a file named female.txt storing name of all the female students.

Solution 4

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:41:03 PM
❯ grep -e '.*,.*,.*,F,.*,.*' students.txt | cut -d ',' -f 2 > female.txt
~/Desktop/cs104/tutorials/tutorial_2 ····· 10:41:22 PM
❯ cat female.txt
Kiara Advani
Rashmika Mandanna
Harmanpreet Kaur
```

To accomplish the task, we do the following:

1. We use grep (with -e) to find all lines where 4th field is 'F'.
2. The regex `.*,.*,.*,F,.*,.*` is matched by all lines where 4th field is 'F' and other fields can be anything denoted by `.*`
3. Finally, we get the 2nd column (i.e. the name) of all female students and redirect it to female.txt file.

Exercise 5

Given the file students.txt, find the number of students in the CSE department.

Solution 5

To accomplish the task, we have two ways:

1. You can use grep with -c option.
2. You can find all CSE students using grep and then use wc -l to find the count.

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:44:32 PM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:50:18 PM
❯ grep -c -e 'CSE$' students.txt
3
~/Desktop/cs104/tutorials/tutorial_2 ····· 02:50:34 PM
❯ grep -e 'CSE$' students.txt | wc -l
       3
```

Exercise 6

Given the file students.txt, find the email id of all students whose name starts with either of A/K/S/R, belonging to 2022 batch and have their ID ending with either 0 or 3.

Solution 6

```
~/Desktop/cs104/tutorials/tutorial_2 ····· 03:20:54 PM
❯ cat students.txt
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
~/Desktop/cs104/tutorials/tutorial_2 ····· 03:20:55 PM
❯ grep -e '[^,]*[03],[AKSR].*,2022,[^,]*' students.txt | cut -d ',' -f 3
kforkavya@cse.iitb.ac.in
sakshamrathi@cse.iitb.ac.in
```

To accomplish the task, we do the following:

1. First we use grep to find all such matches. Carefully look at the regex used for pattern matching.
2. Then we use cut to get the email ids.

Thank You !!!