**CSI Task 1.**

**For all parts of this question, you must show FULL working out in binary.**

1. Choose a number between 35 and 50. This will be number A. Choose another number between

60 and 90. This will be number B. Using 2’s complement 8-bit binary arithmetic, calculate number A – number B

giving your answer in binary and decimal. ***(5 marks)***

*number A = 38*

*number B = 62*

1. *Number A into binary 128,64,32,16,8,4,2,1*

***38 = 00100110 (32 + 4+2+1 = 38)*** *0 0 1 0 0 1 1 0*

*38 / 2 = 19 r 0*

*19 / 2 = 9 r 1*

*9 / 2 = 4 r 1*

*4 / 2 = 2 r 0*

*2 / 2 = 1 r 0*

*1 / 2 = 0 r 1*

1. *Number B into binary then into 2s complement*

***62 = 00111110 (32 + 16 + 8 + 4 + 2 = 62)*** *128,64,32,16,8,4,2,1*

*62 / 2 = 31 r 0* ***0 0 1 1 1 1 1 0***

*31 / 2 = 15 r 1*

*15 / 2 = 7 r 1*

*7 / 2 = 3 r 1*

*3 / 2 = 1 r 1*

*1 /2 = 0 r 1*

*Now make number B needs to be made into 2s complement using 8 bit binary*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | *0* | *1* | *1* | *1* | *1* | *1* | *0* |
| *1* | *1* | *0* | *0* | *0* | *0* | *1* | *0* |
|  |  |  |  |  |  |  | *1* |
| *1* | *1* | *0* | *0* | *0* | *0* | *1* | *0* |

1s

2s



*11000010 = -62* ***(-128 + 64 + 2 = -62)***



1. *Number A – Number B*

*+38 – (-62) = -24*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | *0* | *1* | *0* | *0* | *1* | *1* | *0* |
| *1* | *1* | *0* | *0* | *0* | *0* | *1* | *0* |
|  |  |  |  |  |  |  |  |
| *0* | *0* | *1* | *0* | *1* | *0* | *0* | *0* |

*128 64 32 16 8 4 2 1*

*-32 + 8 = -24*

*Number A – Number B =*

* *24 00101000*

1. Choose one number from the set { 21, 22, 23, 24, 25, 26, 27, 28, 29 } this is number C. Choose a number from { 3, 5, 6 } this is number D. Using 8-bit binary arithmetic, calculate number C × number D

giving your answer in binary and decimal. ***(5 marks)***

*Number C = 25*

*16 + 8 + 1 = 25*

*Binary - 00011001*

*Number D = 5*

*4+1=5*

*Binary - 00000101*

*Number C x Number D*

*128 64 32 16 8 4 2 1*



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***0*** | ***0*** | ***0*** | ***1*** | ***1*** | ***0*** | ***0*** | ***1*** |
| ***0*** | ***0*** | ***0*** | ***0*** | ***0*** | ***1*** | ***0*** | ***1*** |
| *0* | *0* | *0* | *1* | *1* | *0* | *0* | *1* |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *1* | *1* | *0* | *0* | *1* | *0* | *0* |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |

*0 1 1 1 1 1 0 1*

*64 + 32 + 16 + 8 + 4 + 1 =125*

*number c x number d = 125*

*01111101*

*25 x 5 = 125*

1. Using number C and number D from (b), use binary arithmetic to calculate: numberC.375 + numberD.5

giving your answer in binary and decimal.

Convert numberC.375 into IEEE-754 format.

***(5 marks)***

Number C = 25

Number D = 5

Number 25.375 + Number 5.5 = 30.875

-Number 25.375

25/2 = 12 R 1 0.375 X 2 = 0.75 0

12/2 = 6 R 0 0.75 X 2 = 1.500 1

6/2 = 3 R 0 0.5 X 2 =1.000 1

3/2 = 1 R 1

1/2 = 0 R 1 0.375 = 0.011

25 = 11001

25.375 = 11001.011

Now turn this into IEEE-754

11001.011 into 1.1001011

1.1001011 x 2^4

S = 0

E= 4 = 116 = 110100

M= 1001011 ( 23 BITS)

Total = 0 111010010010111001011100101110 (23 times) this this the conversion

Now Number 25.375 + Number 5.5 = 30.875 = 11110.111

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1. | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1. | 1 | 0 |  |

1 1 1 1 0. 1 1 1

1. Choose a number between 175 and 240 – this is number E. Convert number E into hexadecimal. Choose a number between 140 and 160 – this is number F. Convert number F into octal. Using 8bit binary, calculate number E AND number F

Give your answer in binary, octal, decimal and hexadecimal.

***(5 marks)***

*Number E = 180*

*Number F = 160*

*Number E into Hexadecimal*

*180/16 = 11 R 4 18010 = 0XB4 16*

*11 /16 = 0 R 11*

*A = 10, B = 11, C= 12, D=13, E=14, F=15*

*Answer = B4*

*Binary = 10110100*

*Number F into Octal*

*160/8= 20 R 0*

*20/8= 2.5 = 0.5 X 8 = 4 – 2 R 4*

*2/8= 0.25 = 0 R (0.25 X 8 =2) 2*

*OCTAL = 240 8*

*Binary = 10100000*

Number E AND Number F CALCULATION

*10110100 AND 10100000 = 10011110*

*Decimal NUMBER OF E AND F = 158*

*BINARY = 10011110*

*Octal = 236*

*Hexadecimal=9E*

**CSI Task 2.**

1. Choose a short phrase of between 50-60 characters. You could choose a website headline, song title, etc. Convert this phrase using Caesar shift encryption and a key of 5 to create your ciphertext.

Carry out a brute force attack to try to work out the decryption key and plaintext.

***(10 marks)*** (Make sure you show all of your working out and state any assumptions you make. All of the marks are awarded for demonstrating an understanding of the general process of carrying out the attack on Caesar encryption. This means that no marks will be awarded if your security attack is not brute force, even if you get the correct key and the correct plaintext – as you already know them.)

1. Computer A sends 5 packets of data to computer B using Sliding Windows Flow Control
   * The transmission time (time to put on the network) for a packet of data is 1 'time units'
   * Transmission time for an acknowledgement is 0 ‘time units’ (they are very small)
   * The propagation time (time to travel through network) for any transmission is random (between 2 and 3 ‘time units’, you choose a random time for each packet and acknowledgement sent).
   * B's packet processing time is 1. B cannot process multiple packets simultaneously. •
   * The initial window size is 3

Draw a diagram to show how flow will be controlled while the data is being sent.

1. ***Brute Attack***

***For Key = 1, the cyphertext decrypts to “Life is a test so before the night pass. Get right, get right” Not correct plain text.***

***For Key = 2, the cyphertext decrypts to “mjgf js b uftu tp cfgpst uif ojhiu qbtt” Not correct plain text.***

***For Key = 3, the cyphertext decrypts to “nkhg rt c vguv uq dghqtu vjg pkijv rcuu” Not correct plain text.***

***For Key = 4, the cyphertext decrypts to “ojih lu d whvw vr ehiruv wkh qijkw sdvv” Not correct plain text.***

***For Key = 5, the cyphertext decrypts to “pmji mv e xiwx ws fijsvw xii rjkx tww” gre correct plain text.***

**Linux Task 1: Files and Directories (12 marks)**

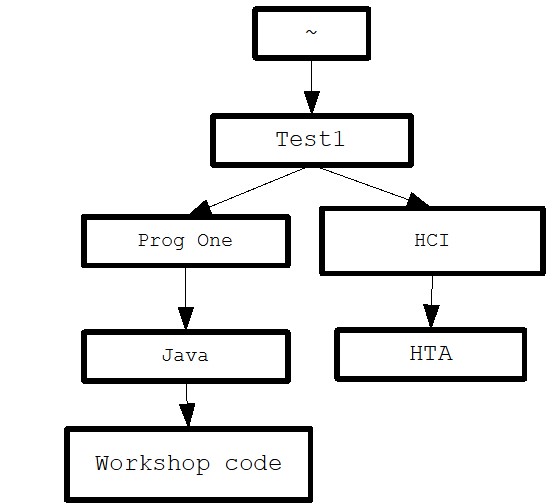
For the following tasks, you can take images of the screen as you perform the tasks and copy those to your assignment document.

Create a directory called “~/Test1” and make it your current working directory.

**Task 1.1 (6 marks):** Create the following hierarchy of directories/subdirectories and direct the output of the tree command to a file.

Text

Description automatically generated



**Task 1.2 (3 marks):** Use vi to make a file called “HTA.txt” inside the “Workshop code” directory. The content of the file should include your full name and group number.

Text

Description automatically generated

Use nano to make a file called “code.txt” inside the “HTA” directory. The content of the file should include your favourite movie.

A picture containing text

Description automatically generated

Use cat to output the contents of your files to the screen and add to your assignment output.

Use tree to show that your files are in the correct location and add to your assignment output.

**Task 1.3 (3 marks):** Using absolute filenames (ones that start with a /), move the HTA.txt file into the “HTA” directory

cd into the “Java” directory. Using relative filenames (ones that *don't* start with a /), copy the code.txt file into the “Workshop code” directory

cd into the “HTA” directory. Using relative filenames (.. notation), rename the “Prog One” directory so it is called “Programming 1”

Shape, rectangle

Description automatically generated

**Text

Description automatically generated**

Text

Description automatically generated

Text

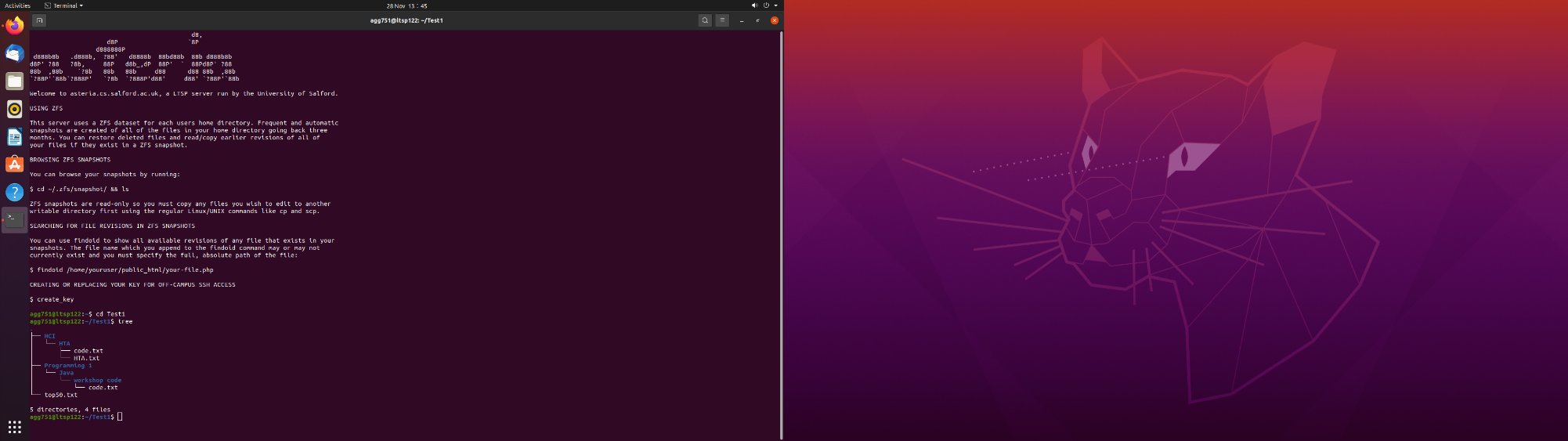
Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated



Run the tree command. Your final output should also include your command history. Text

Description automatically generated

**Linux Task 2 & 3 grep and sed (48 marks):**

Download a list of the top selling songs (top50.txt) details from Blackboard where this assignment was located.

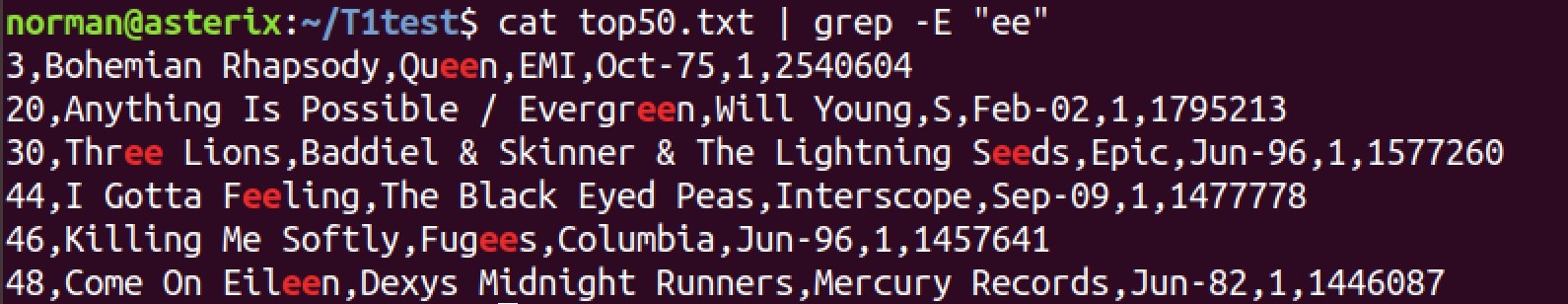
You can now apply regular expressions to the top50.txt file. Look carefully at this file to make sure you understand its format (and that you downloaded it correctly.)

For each of the problems below, you need to use grep (do not use cut or awk) to develop the actual regular expression.

For your grep answers you need to include the command and the output of the command when it is run. You can do this by running the query and taking an image. For example:

Question:

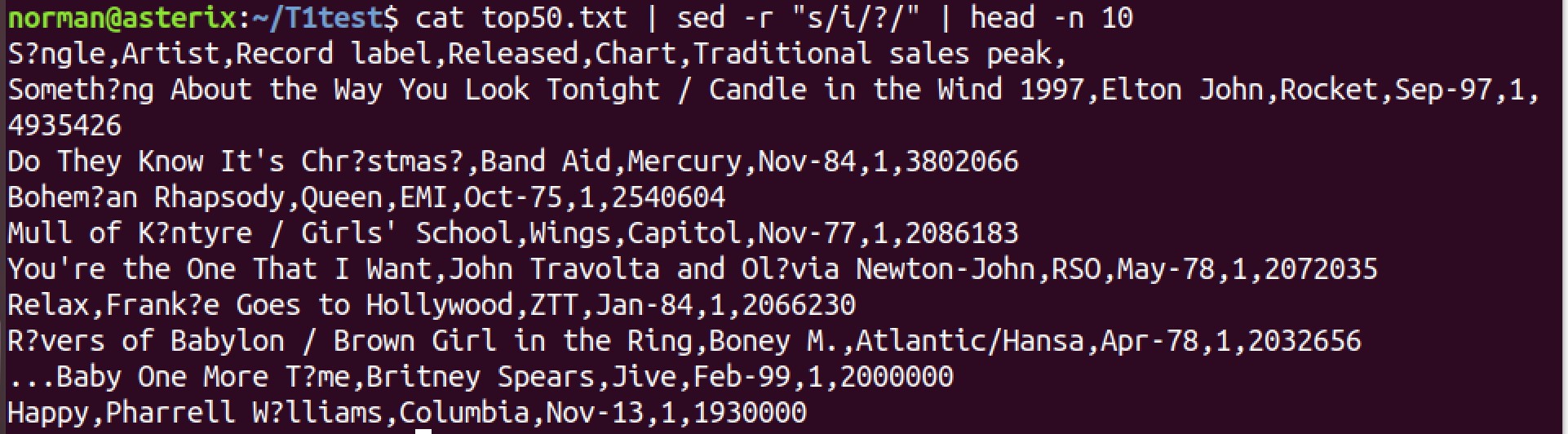
2.0 Find lines with occurrences of double ‘e’.



For the sed problems below, you need to include the sed command and the first 10 lines of the output of the sed command. For example:

Question:

* 1. Change the first instance of an i to a ?



**Marking scheme** (questions out of 4) Make sure you have a go at all the questions.

* 0 means “No attempt”
* 1 means “I can see what you are trying to do”
* 2 means “A reasonable approach that shows you're on the right lines”.
* 3 means “Does (or could) match a couple of lines too many or too few, but nearly there” • 4 means “Perfect”

**Task 2 – These questions need to be answered using grep (32 marks – 4 marks per part):**

2.1: What Beatles songs are in the top 50?

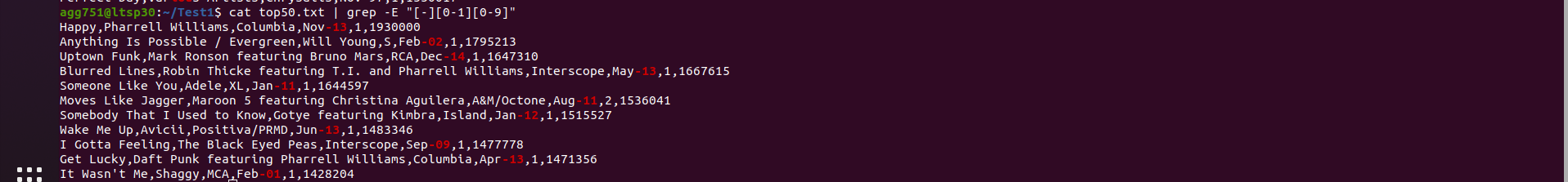
Text

Description automatically generated

2.2: Find all lines with words with 3 consecutive vowels.



2.3: Find all the hits from the 2000’s and the 2010’s.

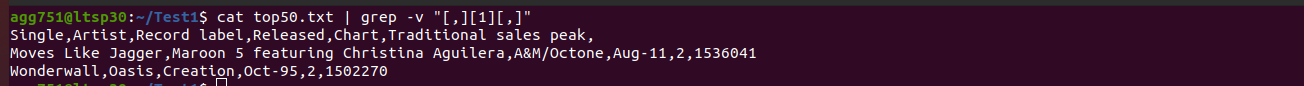


2.4: Find all songs with one-word titles.

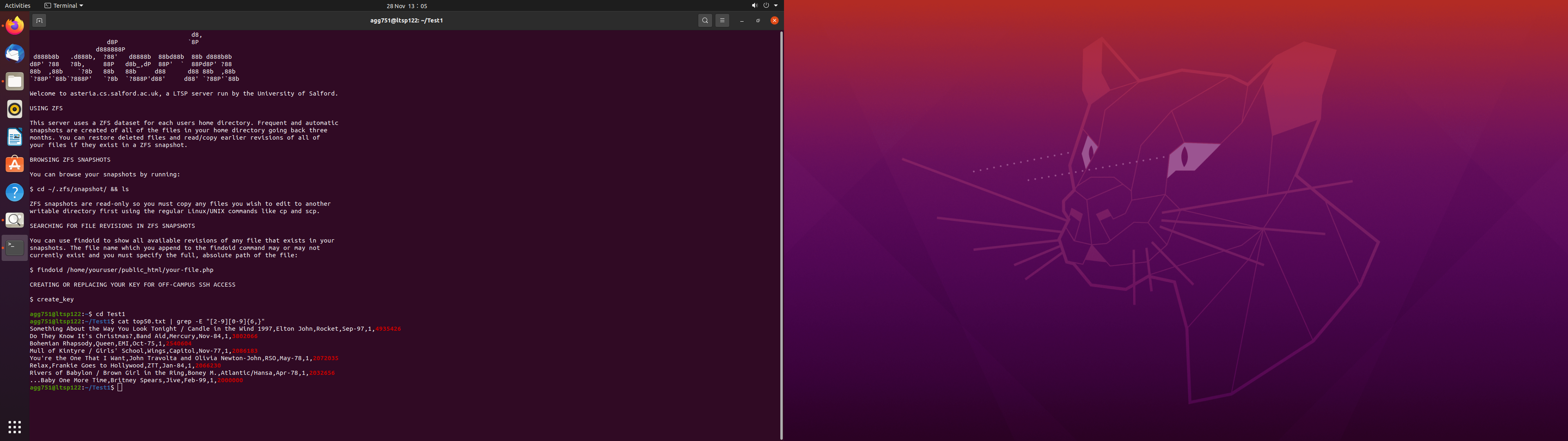


2.5: Find all songs whose name is at least 35 characters. 

2.6: What songs did not make it to number 1?



2.7: Find the songs with sales of 2 million or more.



2.8: What year has the most hit songs in the top 50?

Text

Description automatically generated

**Task 3 – sed (16 marks – 4 marks per part):**

**Please do the questions with 'stream-based editing'** (cat filename | sed -r …)

**Output needs to include the command and the first 10 lines of output.**

3.1: Change all occurrences of 'Queen' to ‘King’.

A picture containing graphical user interface

Description automatically generated

3.2: Change all spaces to underscores. Text

Description automatically generated

3.3: For the first 5 songs in the file, change the song name to uppercase.

3.4: With lines that contain "Wings" append the line "Above was in The Beatles."

