BINF2111 – Introduction to Bioinformatics Computing

UNIX 101 – enter the coding zone

UNIX

Richard Allen White III, PhD
RAW Lab
Lecture 6 – Thursday Sep 8th, 2021

Learning Objectives

- Review quiz and bonus

- wget/zip/tar

- tr/printf command

- text editors

- Sort/uniq/cut commands review

- Quiz 6

Quiz 5 answers

```
- grep -v -e '^$' file.txt
```

A) delete all empty lines
B) cut all empty lines
C) print all empty files
D) append all empty lines

Quiz 5 answers

- grep -v -e '^$' file.txt

A) delete all empty lines

B) cut all empty lines

C) print all empty files

D) append all empty lines

Quiz 5 answers

- cut -f1,4 -d "," file.csv --complement

A) print only columns 1 and 4
B) print all but columns 1 and 4
C) cut only columns 1 and 4 but not print
D) will email Dr. White and Jose the columns

Quiz 5 answers

- `cut -f1,4 -d "," file.csv --complement`

A) print only columns 1 and 4
B) print all but columns 1 and 4
C) cut only columns 1 and 4 but not print
D) will email Dr. White and Jose the columns

Lab Two most missed question

Count the numbers of 'A', 'T', 'G', 'C' using multiple or single grep commands in example.fasta?

Lab Two most missed question

Count the numbers of 'A', 'T', 'G', 'C' using multiple or single grep commands in example.fasta?

```
egrep -o 'A|T|G|C' example.fasta | wc -l
grep -oE 'A|T|G|C' example.fasta | wc -l
```

168?

Lab Two most missed question

Count the numbers of 'A', 'T', 'G', 'C' using multiple or single grep commands in example.fasta?

```
egrep -o 'A|T|G|C' example.fasta | wc -l
grep -oE 'A|T|G|C' example.fasta | wc -l
```

168?

THIS WOULD BE WRONG! But, why?

Lab Two most missed question

```
>chr1_geneA
CTAAGGCTATCTTGACAACTGACT
>chr1_geneB
CTAAGGCTATGTTGGCAACTGACT
>chr1_geneC
CTAAGGCTACCTTGACAACTGACT
>chr1_geneD
AAAAGGCTATCTTGACAACTGACT
>chr1_geneX
CTAAGGCTATCTTGATTTCTGACT
>chr1_geneY
GGGGGGCTATCTTGACAACTGACT
>chr1_geneZ
CTAAGGCTATCNNGACAACTGACT
```

Lab Two most missed question

```
>chr1_geneA
CTAAGGCTATCTTGACAACTGACT
>chr1_geneB
CTAAGGCTATGTTGGCAACTGACT
>chr1_geneC
CTAAGGCTACCTTGACAACTGACT
>chr1_geneD
AAAAGGCTATCTTGACAACTGACT
>chr1_geneX
CTAAGGCTATCTTGATTTCTGACT
>chr1_geneY
GGGGGGCTATCTTGACAACTGACT
>chr1_geneZ
CTAAGGCTATCNNGACAACTGACT
```

Lab Two most missed question

```
>chr1_geneA
CTAAGGCTATCTTGACAACTGACT
>chr1_geneB
CTAAGGCTATGTTGGCAACTGACT
>chr1_geneC
CTAAGGCTACCTTGACAACTGACT
>chr1_geneD
AAAAGGCTATCTTGACAACTGACT
>chr1_geneX
CTAAGGCTATCTTGATTTCTGACT
>chr1_geneY
GGGGGGCTATCTTGACAACTGACT
>chr1_geneZ
CTAAGGCTATCNNGACAACTGACT
```

Write a command where you could fix this?

Lab Two most missed question

```
>chr1_geneA
CTAAGGCTATCTTGACAACTGACT
>chr1_geneB
CTAAGGCTATGTTGGCAACTGACT
>chr1_geneC
CTAAGGCTACCTTGACAACTGACT
>chr1_geneD
AAAAGGCTATCTTGACAACTGACT
>chr1_geneX
CTAAGGCTATCTTGATTTCTGACT
>chr1_geneY
GGGGGGCTATCTTGACAACTGACT
>chr1_geneZ
CTAAGGCTATCNNGACAACTGACT
```

```
sed 's/geneA/geneX2/g' example.fasta | sed
‘s/geneC/geneX3/g’ | egrep -o 'A|T|C|G' | wc -l
```

Lab Two most missed question

```
>chr1_geneA
CTAAGGCTATCTTGACAACTGACT
>chr1_geneB
CTAAGGCTATGTTGGCAACTGACT
>chr1_geneC
CTAAGGCTACCTTGACAACTGACT
>chr1_geneD
AAAAGGCTATCTTGACAACTGACT
>chr1_geneX
CTAAGGCTATCTTGATTTCTGACT
>chr1_geneY
GGGGGGCTATCTTGACAACTGACT
>chr1_geneZ
CTAAGGCTATCNNGACAACTGACT
```

Always check your data! ;-)

Bonus 4

- Convert name game file (name_game.csv) to tsv with:

→ tr

→ awk

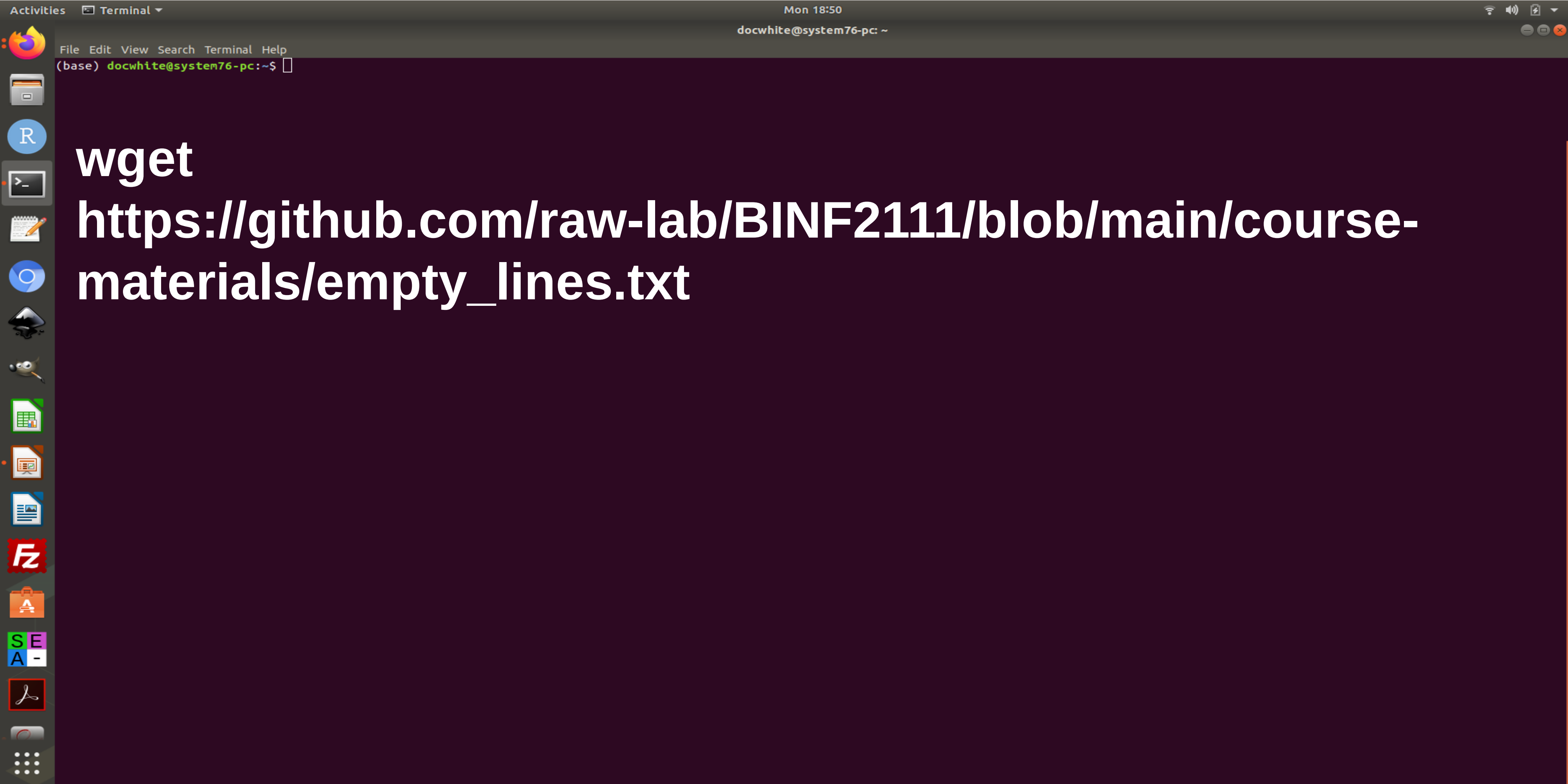
Bonus 4

- Convert name game file (name_game.csv) to tsv with:

→ **tr:** cat name_game.csv | tr -s ',' '\t' >name_game.tsv

→ **awk:** cat name_game.csv | awk -F ',' '{$1=$1}1' >name_game.tsv

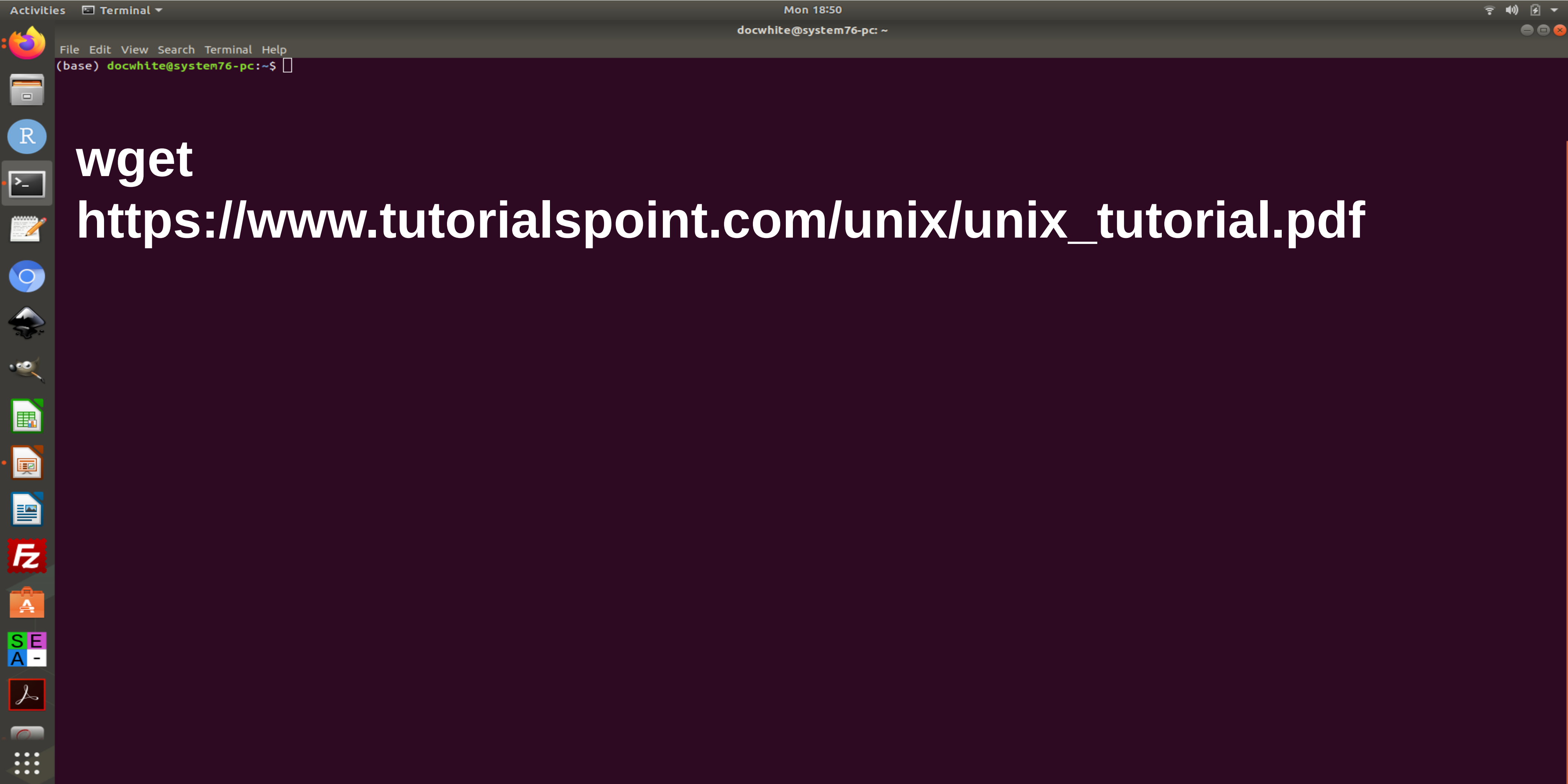
wget examples



```
(base) docwhite@system76-pc:~$
```

```
wget https://github.com/raw-lab/BINF2111/blob/main/course-materials/empty_lines.txt
```

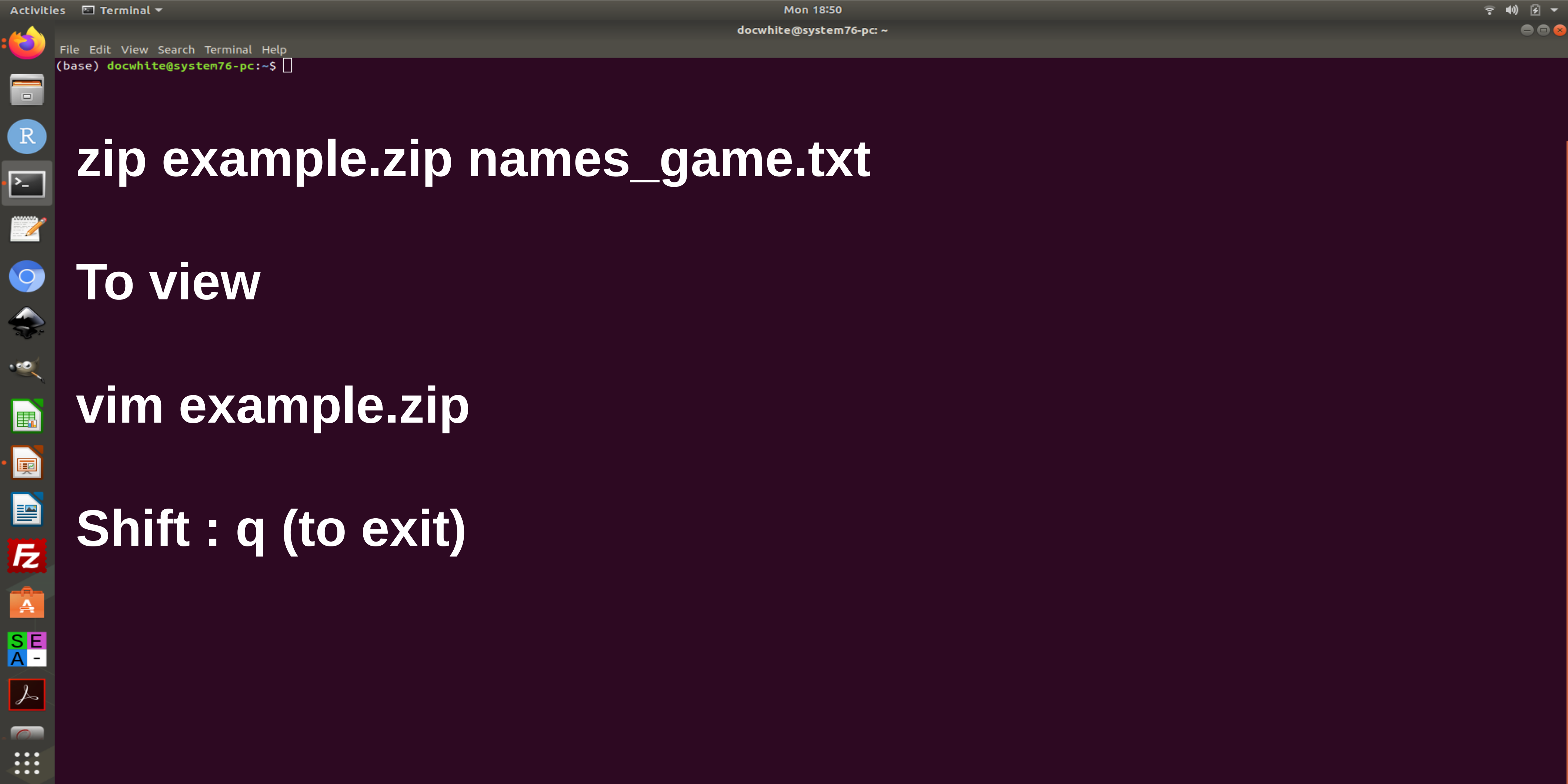
wget examples



```
(base) docwhite@system76-pc:~$
```

```
wget
https://www.tutorialspoint.com/unix/unix_tutorial.pdf
```

zip examples



```
(base) docwhite@system76-pc:~$
```

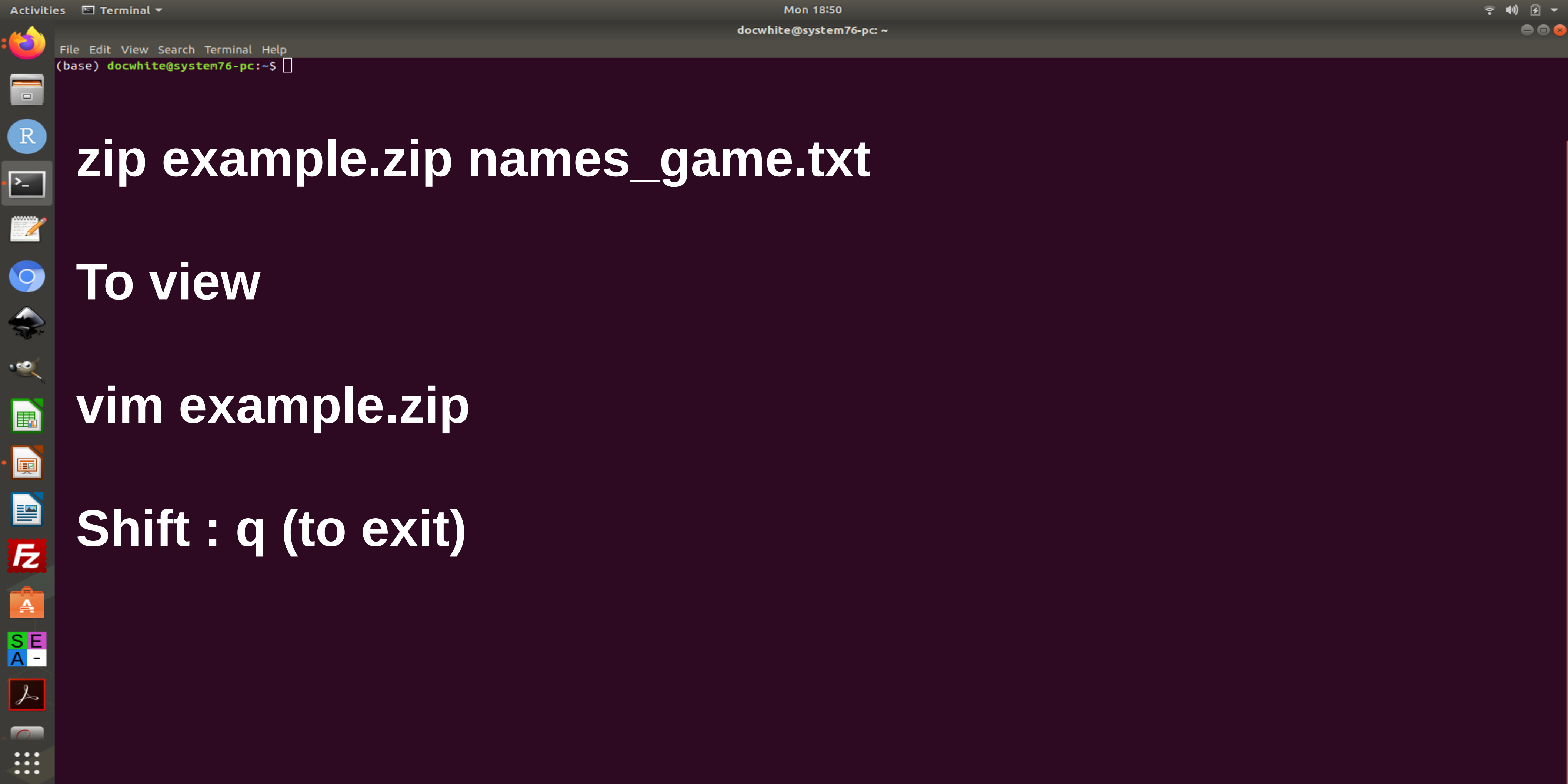
zip example.zip names_game.txt

To view

vim example.zip

Shift : q (to exit)

zip examples



```
(base) docwhite@system76-pc:~$ 
```

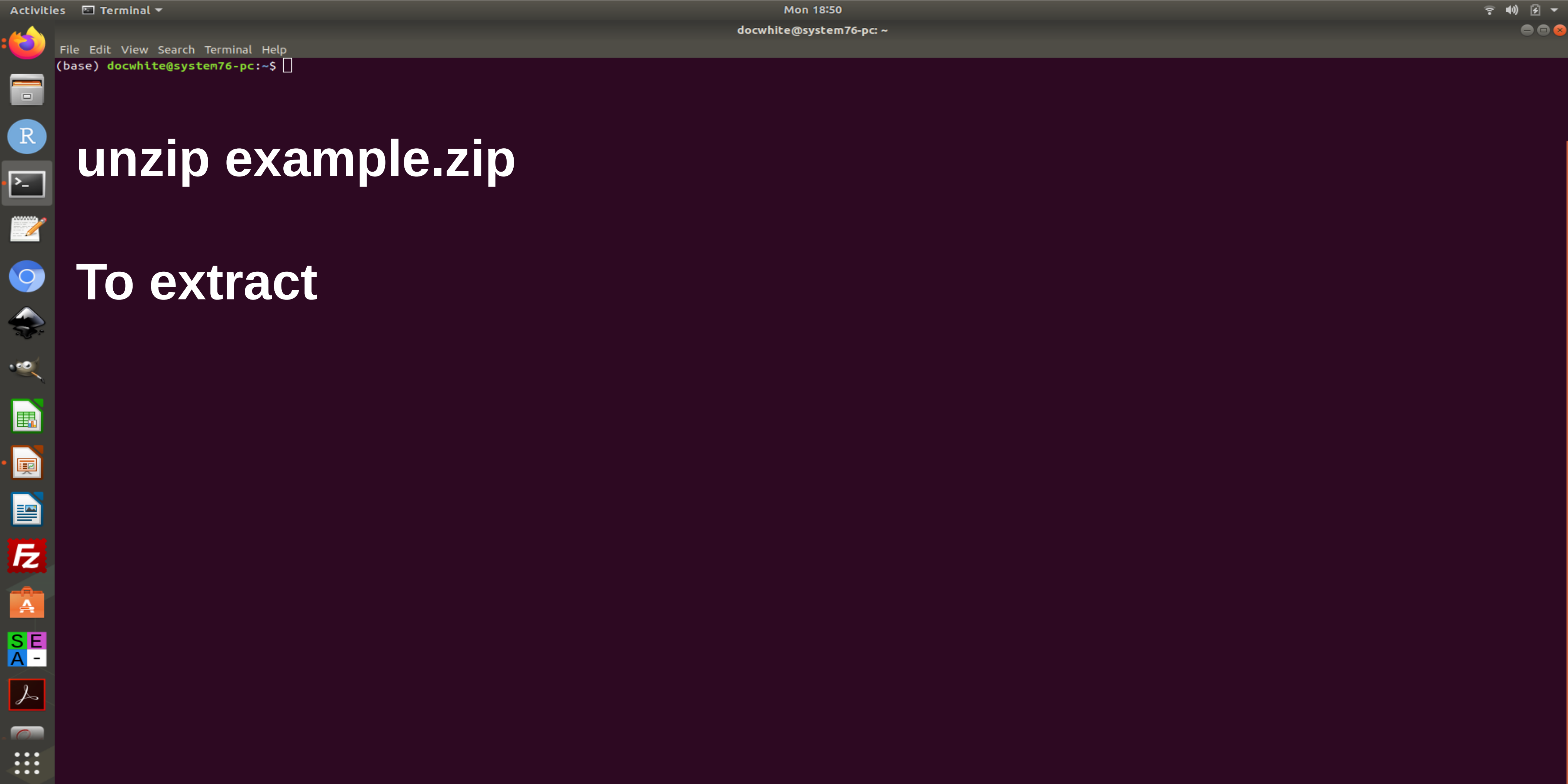
```
zip example.zip names_game.txt
```

To view

```
vim example.zip
```

Shift : q (to exit)

zip examples

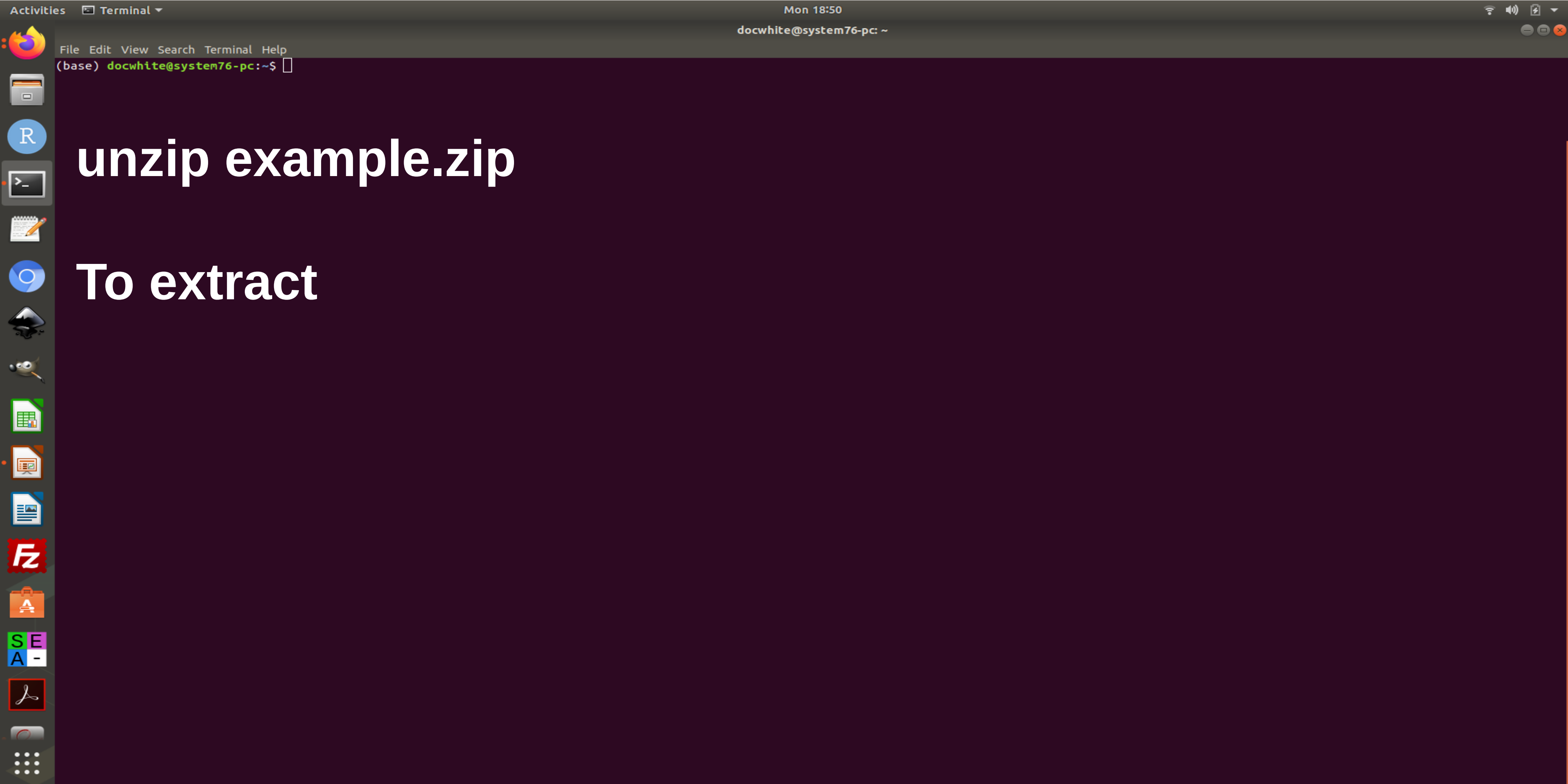


```
(base) docwhite@system76-pc:~$
```

unzip example.zip

To extract

zip examples

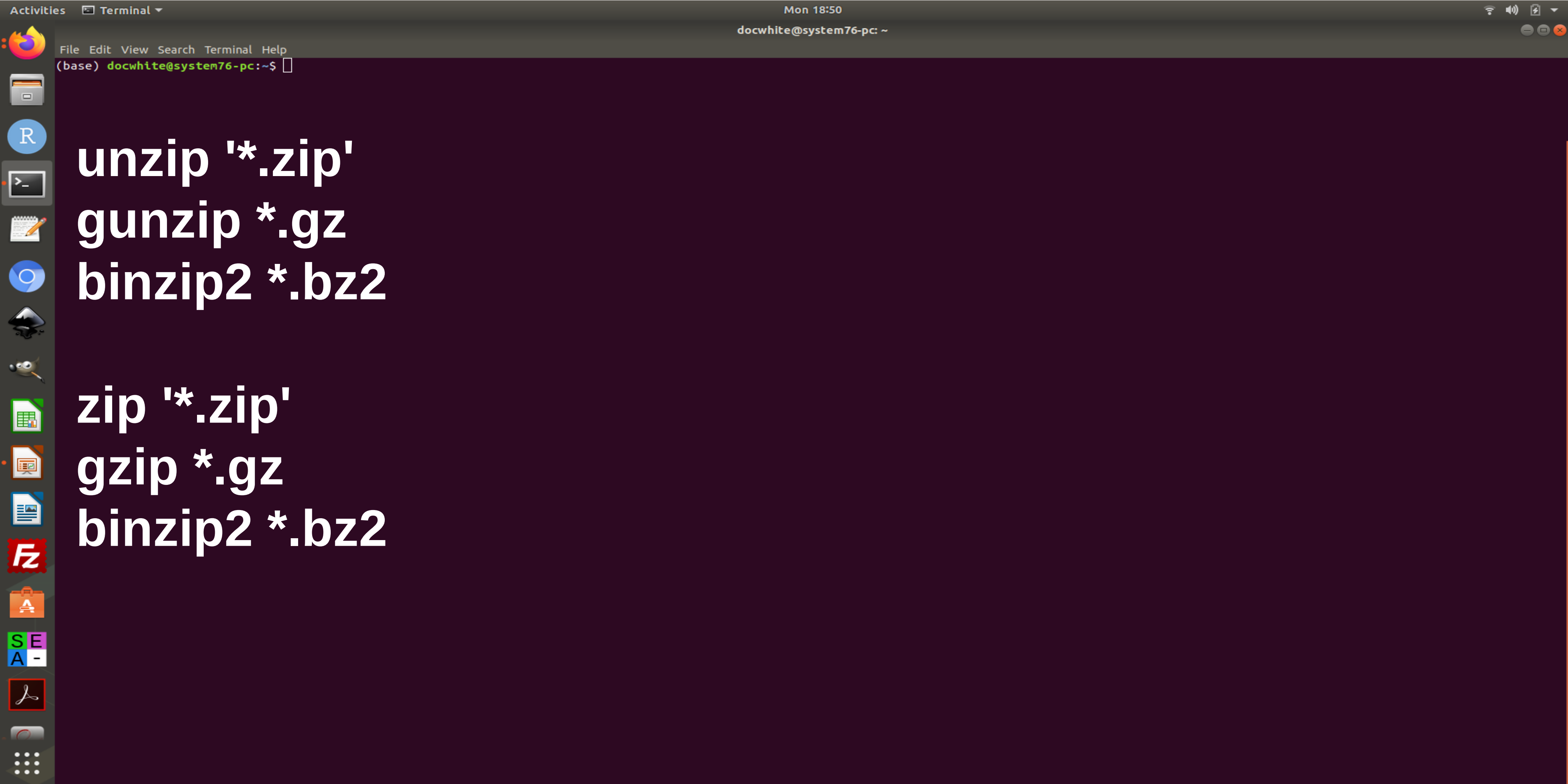


```
(base) docwhite@system76-pc:~$
```

unzip example.zip

To extract

zip examples

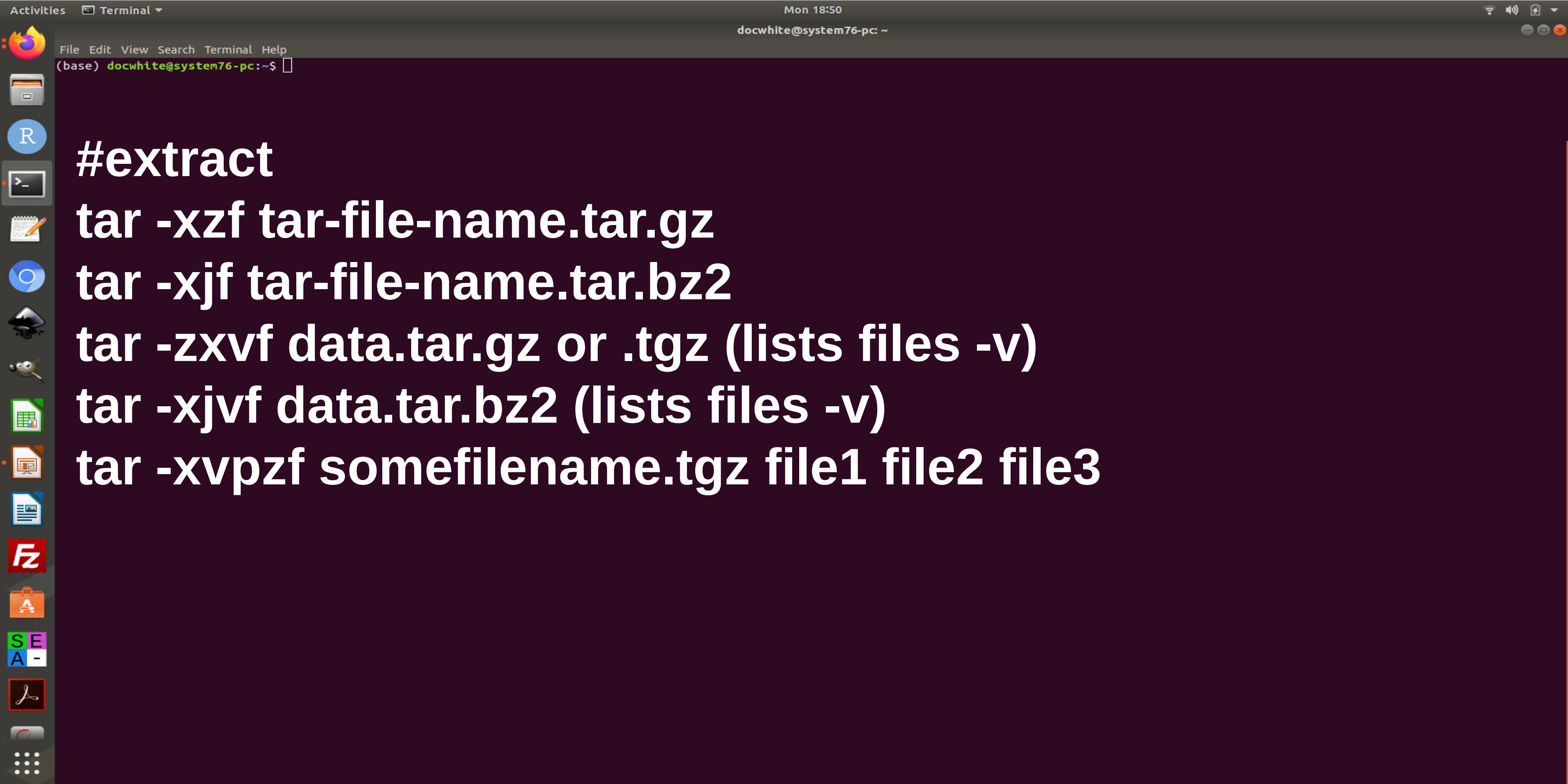


```
(base) docwhite@system76-pc:~$ 
```

```
unzip '*.zip'
gunzip *.gz
binzip2 *.bz2

zip '*.zip'
gzip *.gz
binzip2 *.bz2
```

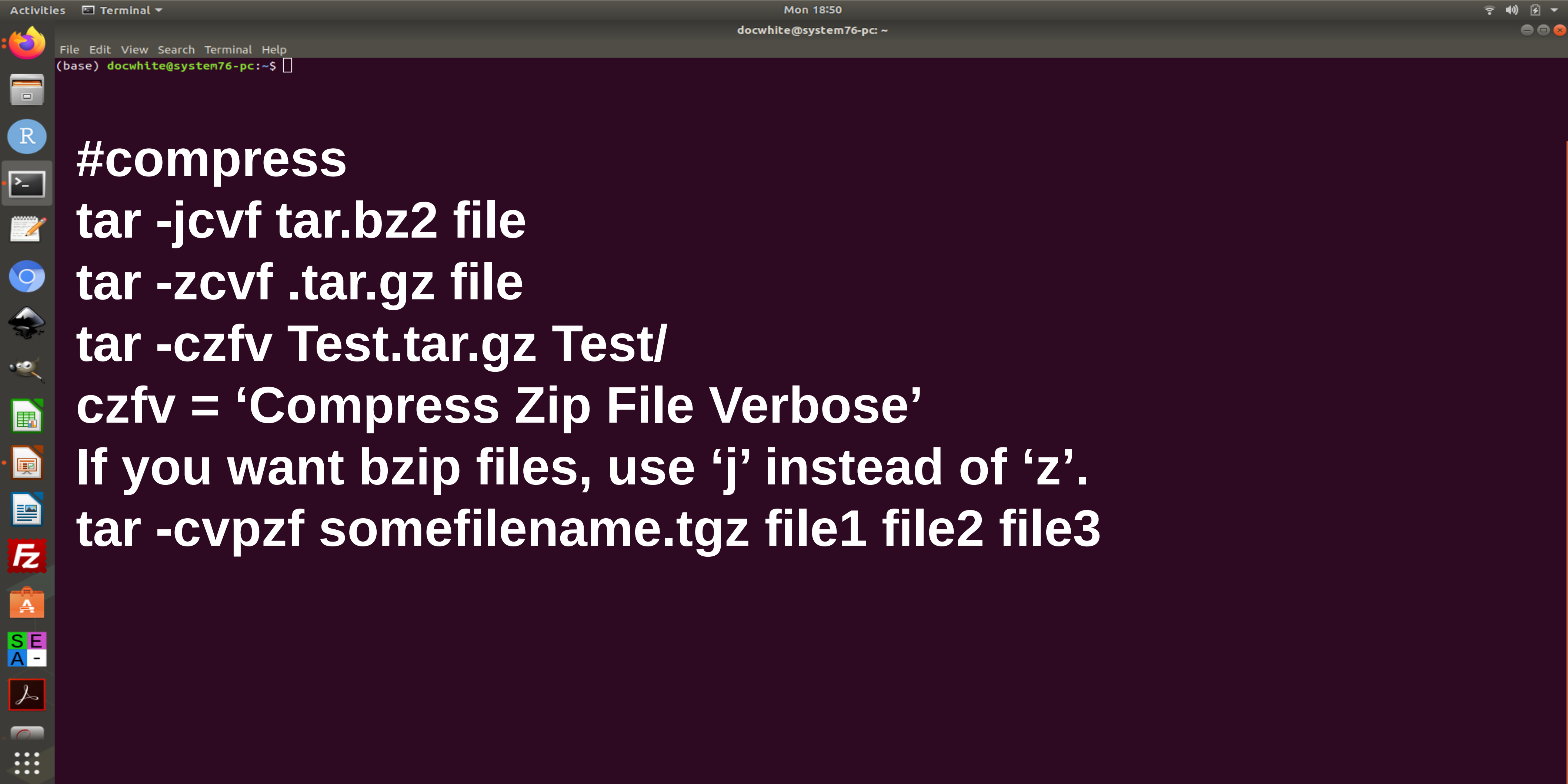
tar examples



```
(base) docwhite@system76-pc:~$
```

```
#extract
tar -xzf tar-file-name.tar.gz
tar -xjf tar-file-name.tar.bz2
tar -zxvf data.tar.gz or .tgz (lists files -v)
tar -xjvf data.tar.bz2 (lists files -v)
tar -xvpzf somefilename.tgz file1 file2 file3
```

tar examples



```
(base) docwhite@system76-pc:~$
```

#compress

tar -jcvf tar.bz2 file

tar -zcvf .tar.gz file

tar -czfv Test.tar.gz Test/

czfv = ‘Compress Zip File Verbose’

If you want bzip files, use ‘j’ instead of ‘z’.

tar -cvpzf somefilename.tgz file1 file2 file3

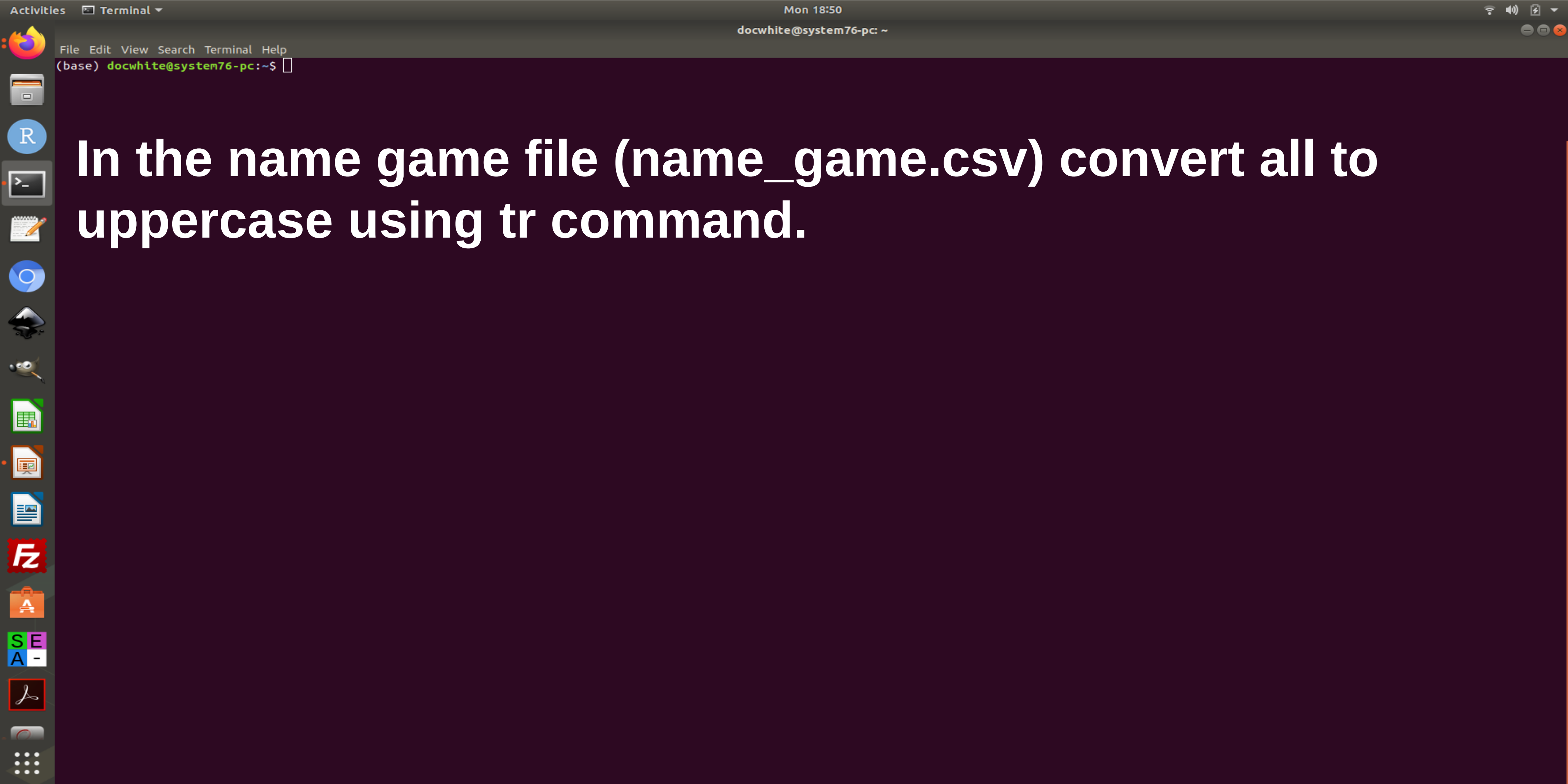
Tr (translate) – syntax anatomy UNIX tool

tr [OPTION] SET1 [SET2]

-c : complements the set of characters in string
(i.e., operations apply to characters not in the given set)
-d : delete characters in the first set from the output.
-s : replaces repeated characters listed in the set1 with single occurrence
-t : truncates set1

tr (no option) = substitute [original] [new]

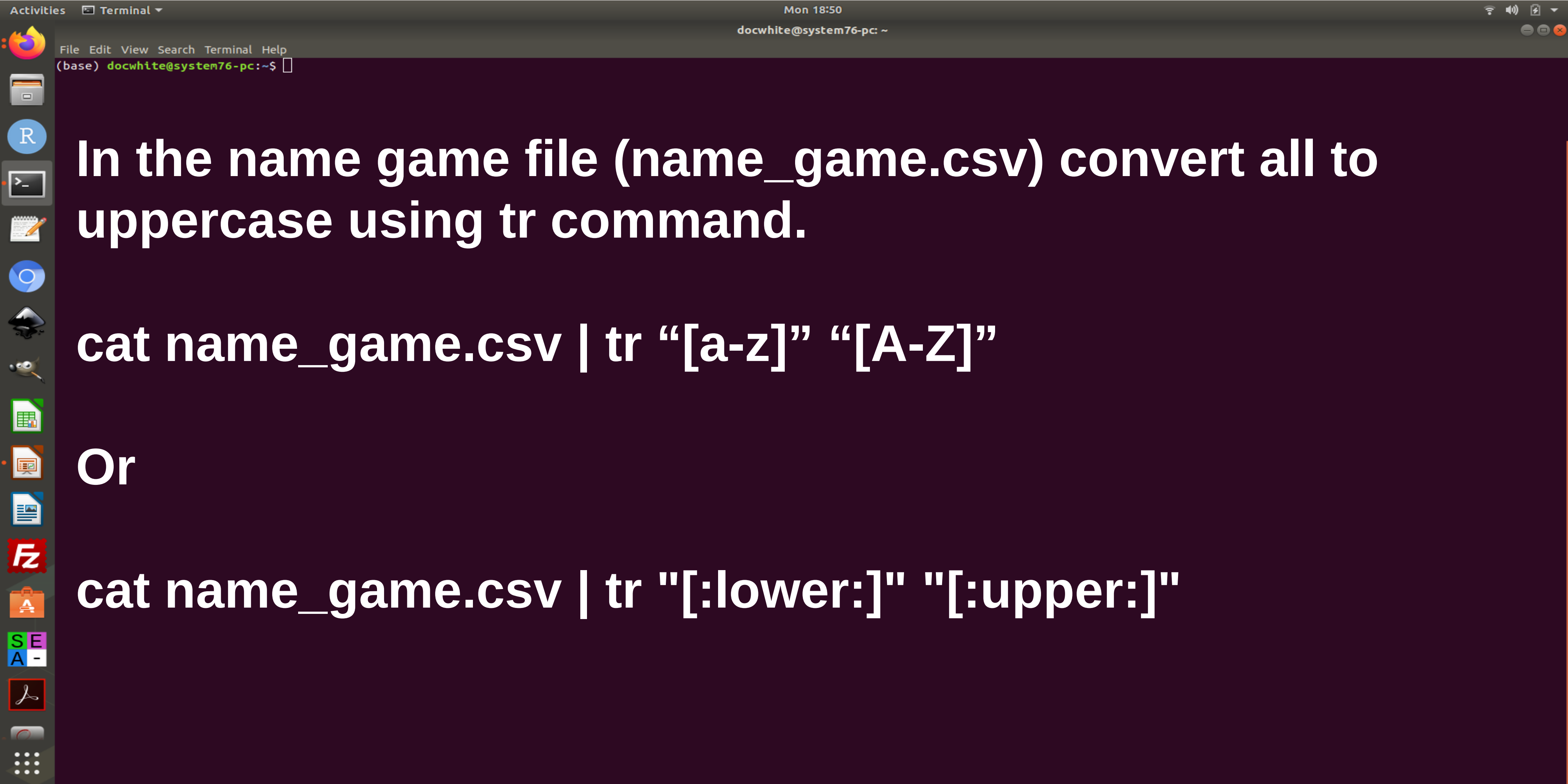
tr examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.csv) convert all to uppercase using tr command.

tr examples



```
(base) docwhite@system76-pc:~$ 
```

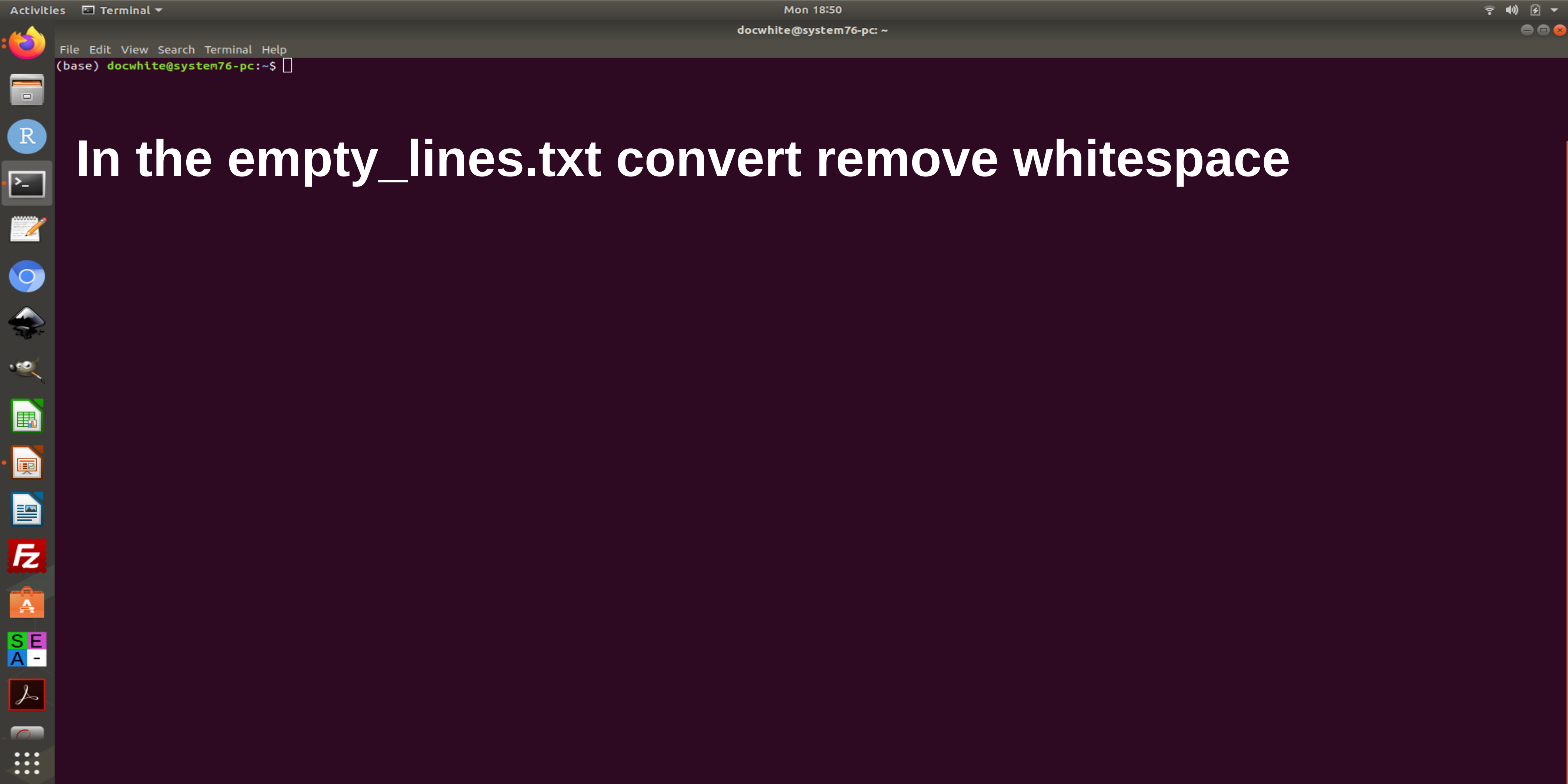
In the name game file (name_game.csv) convert all to uppercase using tr command.

cat name_game.csv | tr “[a-z]” “[A-Z]”

Or

cat name_game.csv | tr "[:lower:]" "[:upper:]"

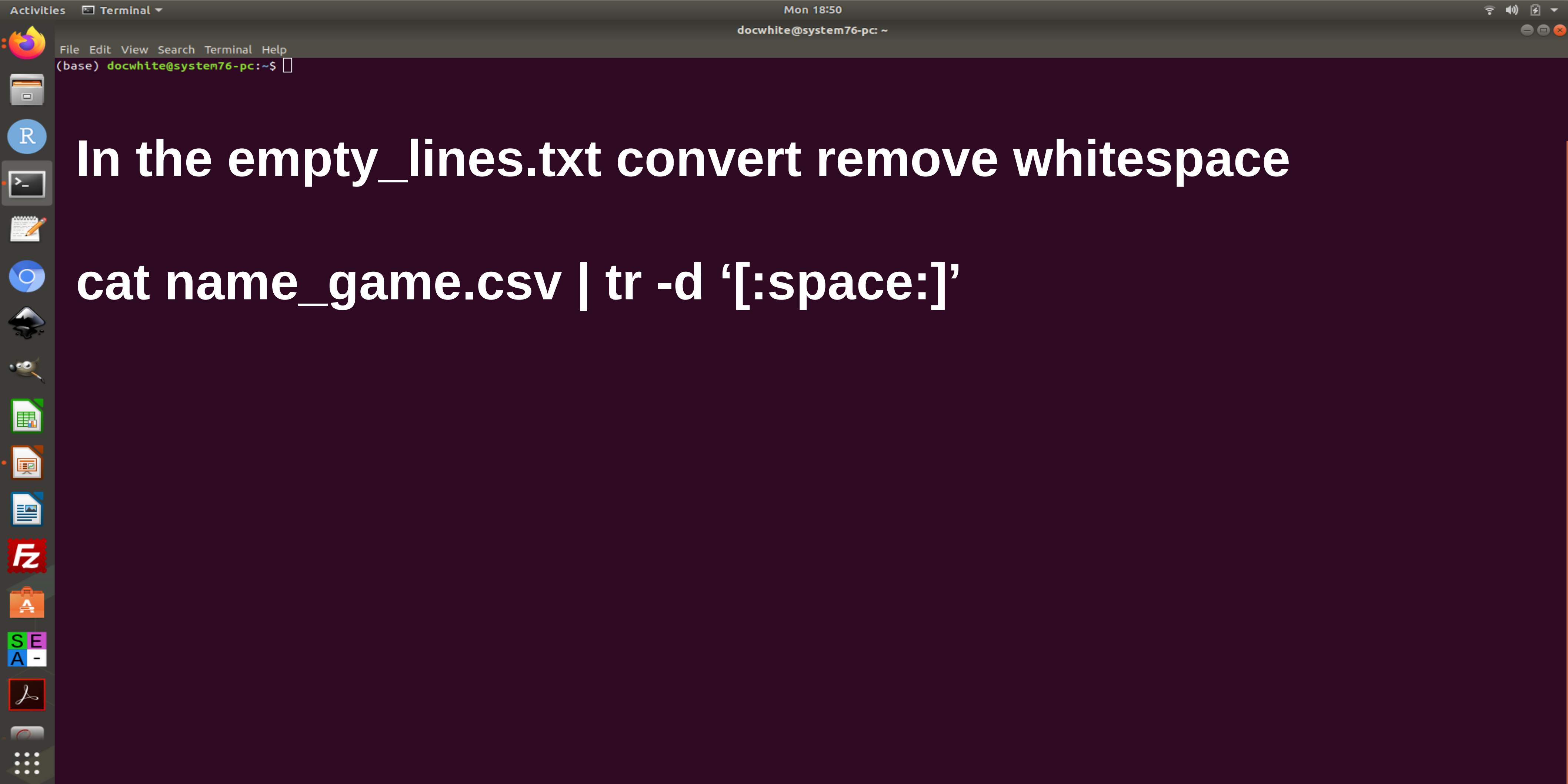
tr examples



```
(base) docwhite@system76-pc:~$ 
```

In the empty_lines.txt convert remove whitespace

tr examples

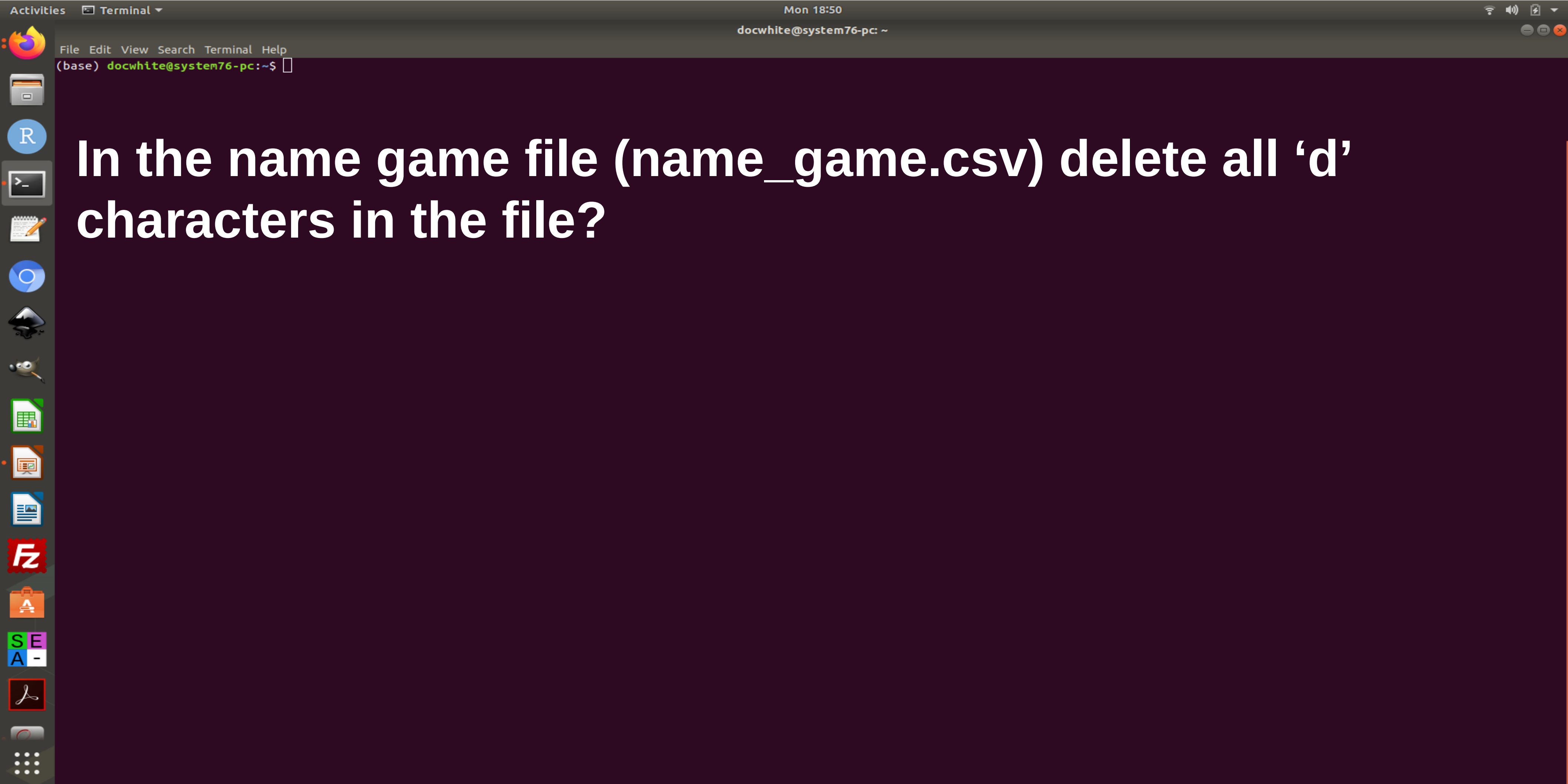


```
(base) docwhite@system76-pc:~$ 
```

In the empty_lines.txt convert remove whitespace

```
cat name_game.csv | tr -d ‘[:space:]’
```

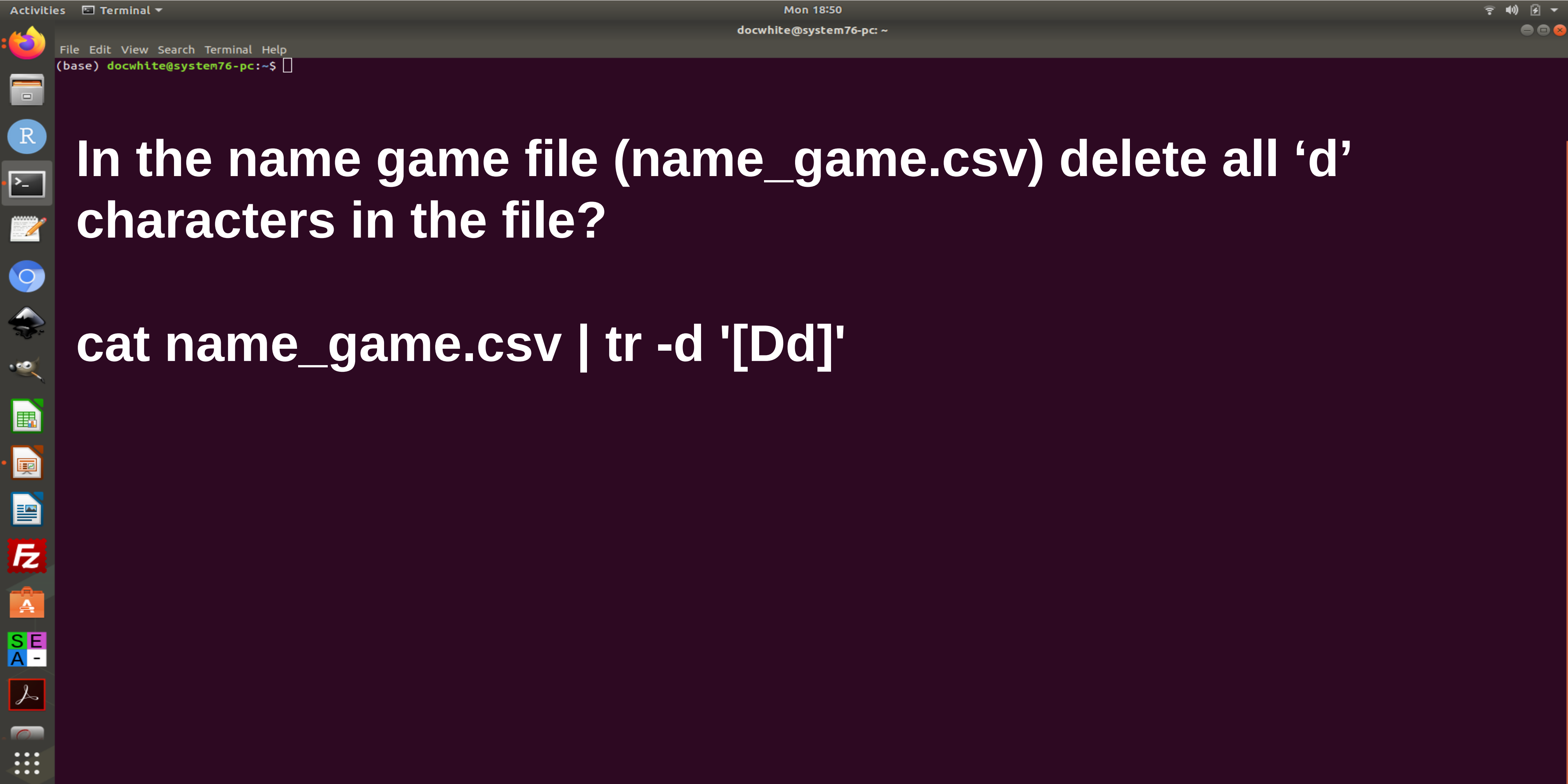
tr examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.csv) delete all ‘d’ characters in the file?

tr examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.csv) delete all ‘d’ characters in the file?

cat name_game.csv | tr -d '[Dd]'

printf – syntax anatomy UNIX

printf [-v var] format [arguments]

%d	For signed decimal numbers
%i	For signed decimal numbers
%u	For unsigned decimal numbers
%o	For unsigned octal numbers
%x	For unsigned hexadecimal numbers with lower case letters (a-f)
%X	For unsigned hexadecimal numbers with upper case letters (A-F)
%f	For floating point numbers
%s	For string
%%	For percent % symbol

printf – syntax anatomy UNIX

printf [-v var] format [arguments]

--help display this help and exit
--version output version information and exit

\xHH byte with hexadecimal value HH (1 to 2 digits)
\uHHHH Unicode (ISO/IEC 10646) character with hex value HHHH (4 digits)
\UHHHHHHHH
Unicode character with hex value HHHHHHHH (8 digits)
%% a single %
%b ARGUMENT as a string with '\' escapes interpreted, except that octal escapes are of the form \0 or \0NNN

\" double quote
\NNN character with octal value NNN (1 to 3 digits)
\\ backslash
\a alert (BEL)
\b backspace
\c produce no further output
\f form feed
\n new line
\r carriage return
\t horizontal tab
\v vertical tab

printf – syntax anatomy UNIX

printf [-v var] format [arguments]

N This specifies the width of the field for output.

* This is the placeholder for the width.

- To left align output in the field. (Default: Right align)

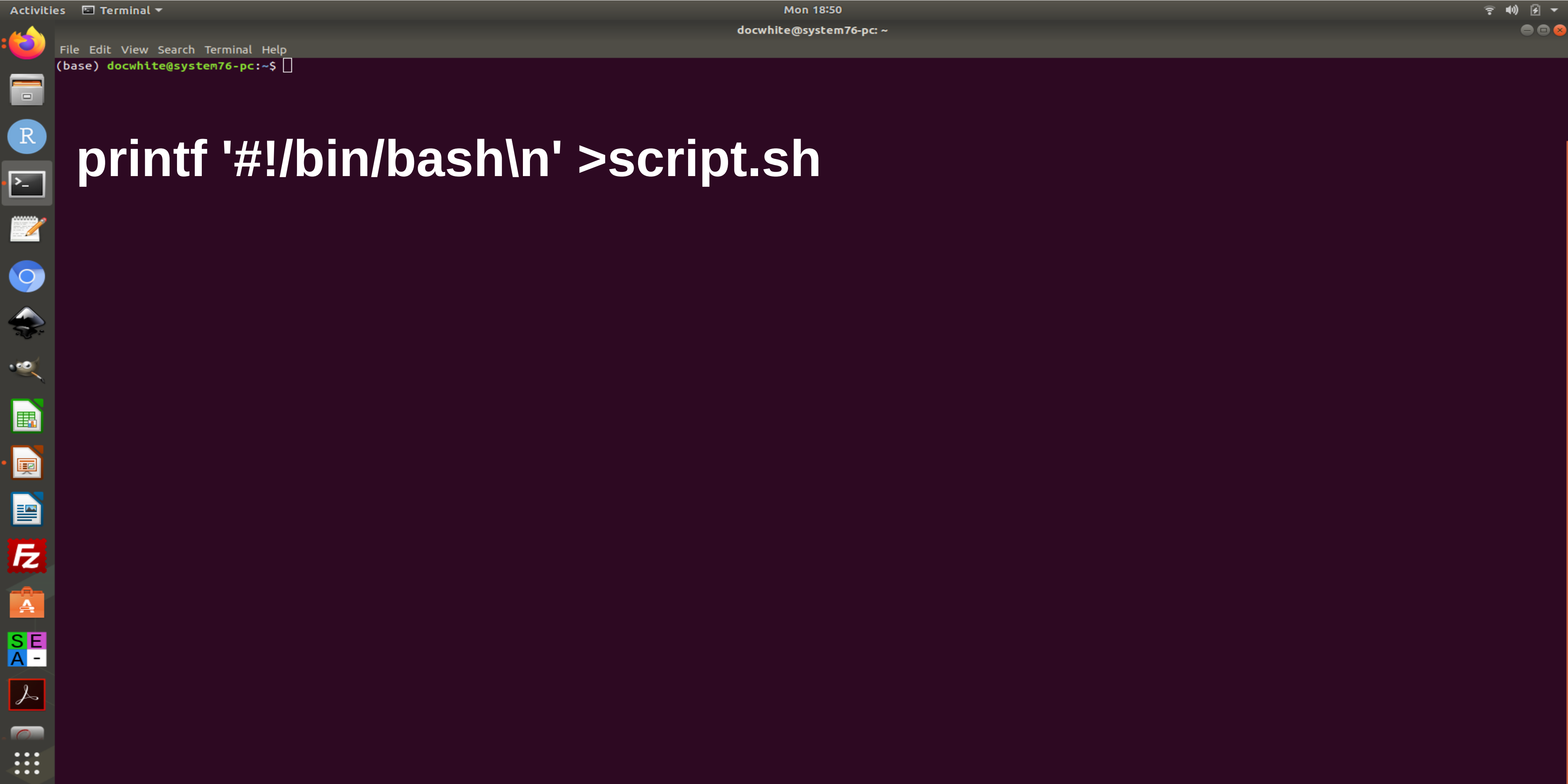
0 Pad result with leading 0s.

+ To put + sign before positive numbers and - sign for negative numbers.

printf() function of C programming language.

We can say that printf is a successor of echo command.

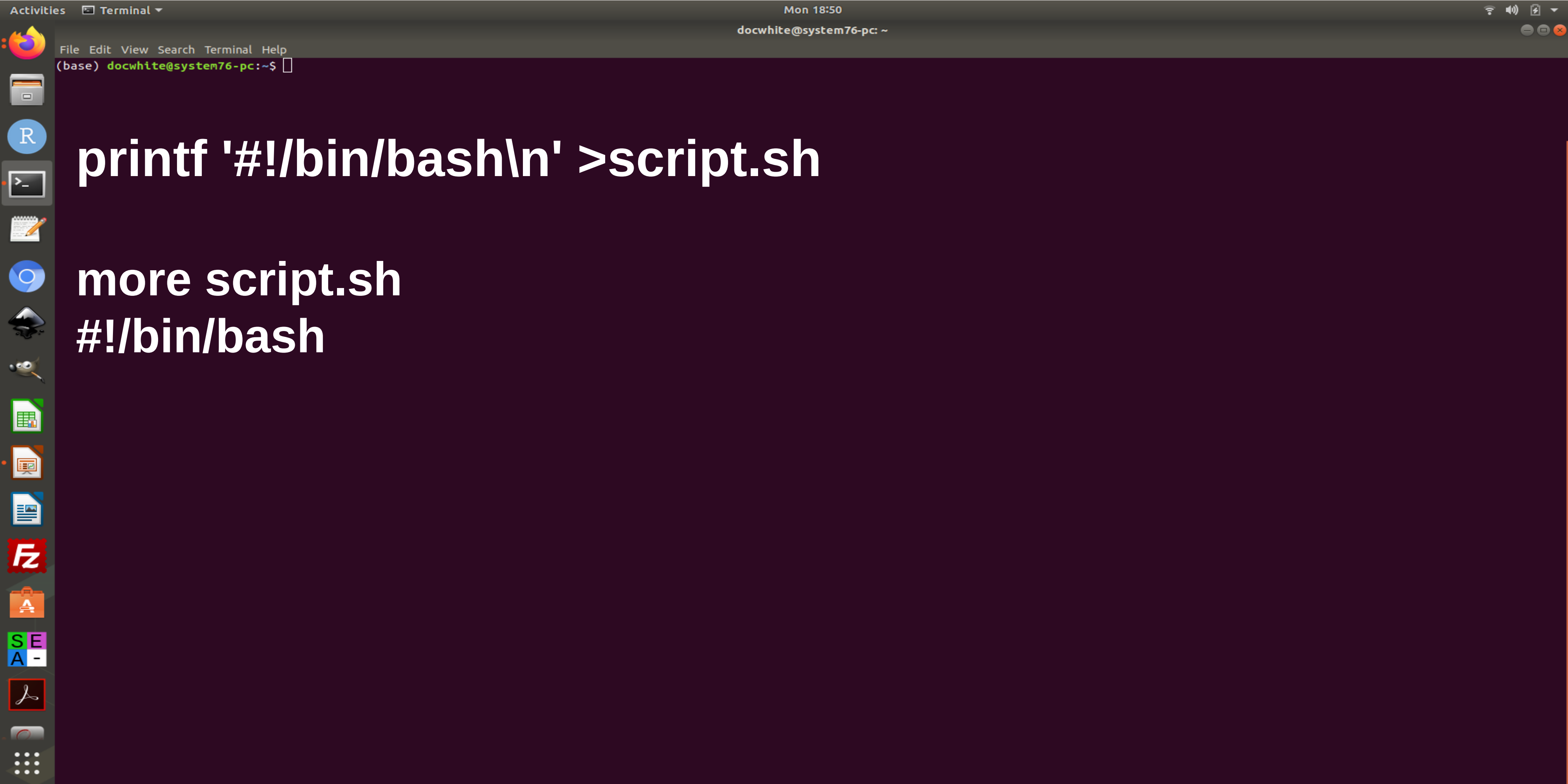
printf examples



```
(base) docwhite@system76-pc:~$ 
```

```
printf '#!/bin/bash\n' >script.sh
```

printf examples

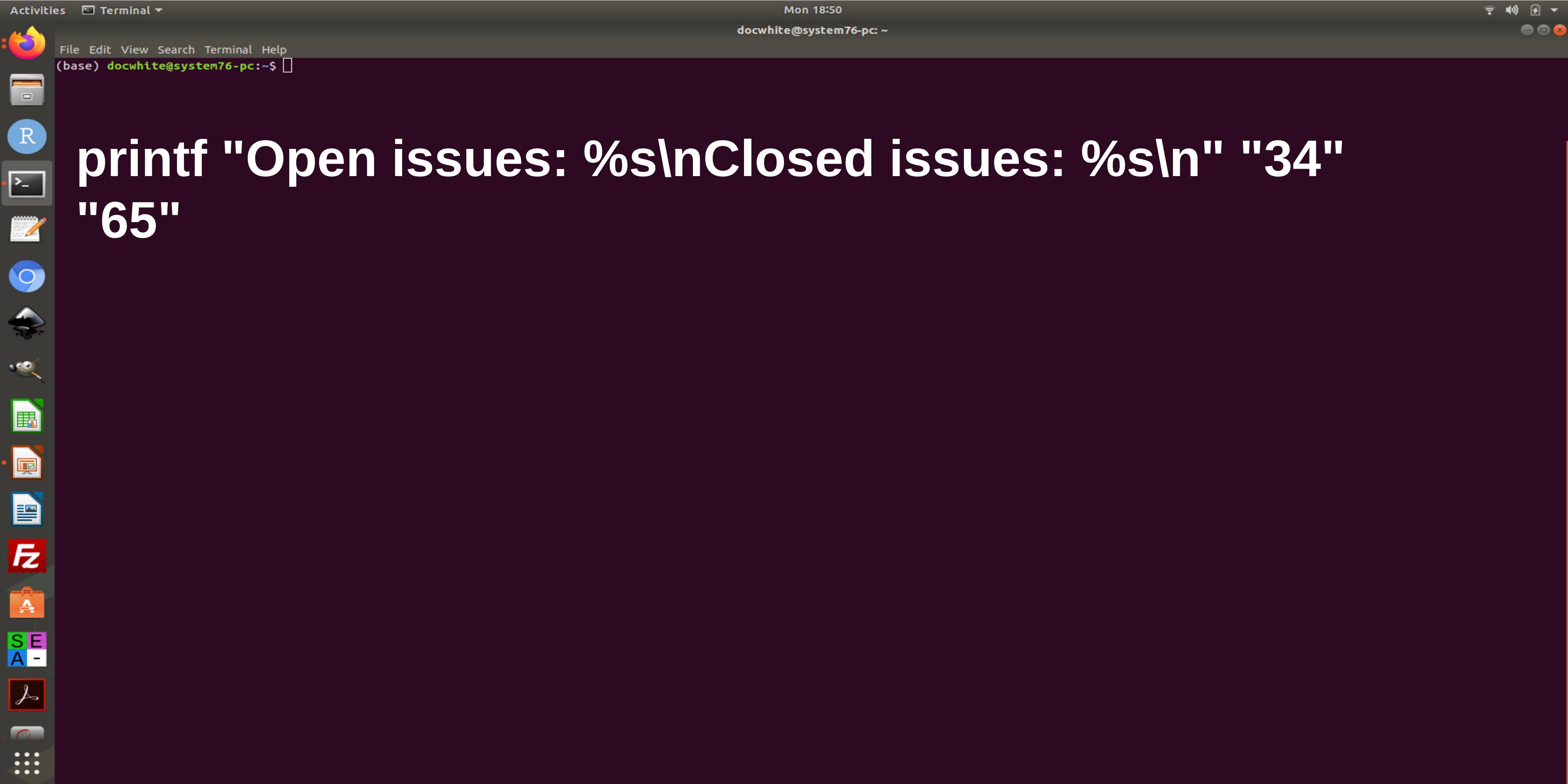


```
(base) docwhite@system76-pc:~$
```

```
printf '#!/bin/bash\n' >script.sh

more script.sh
#!/bin/bash
```

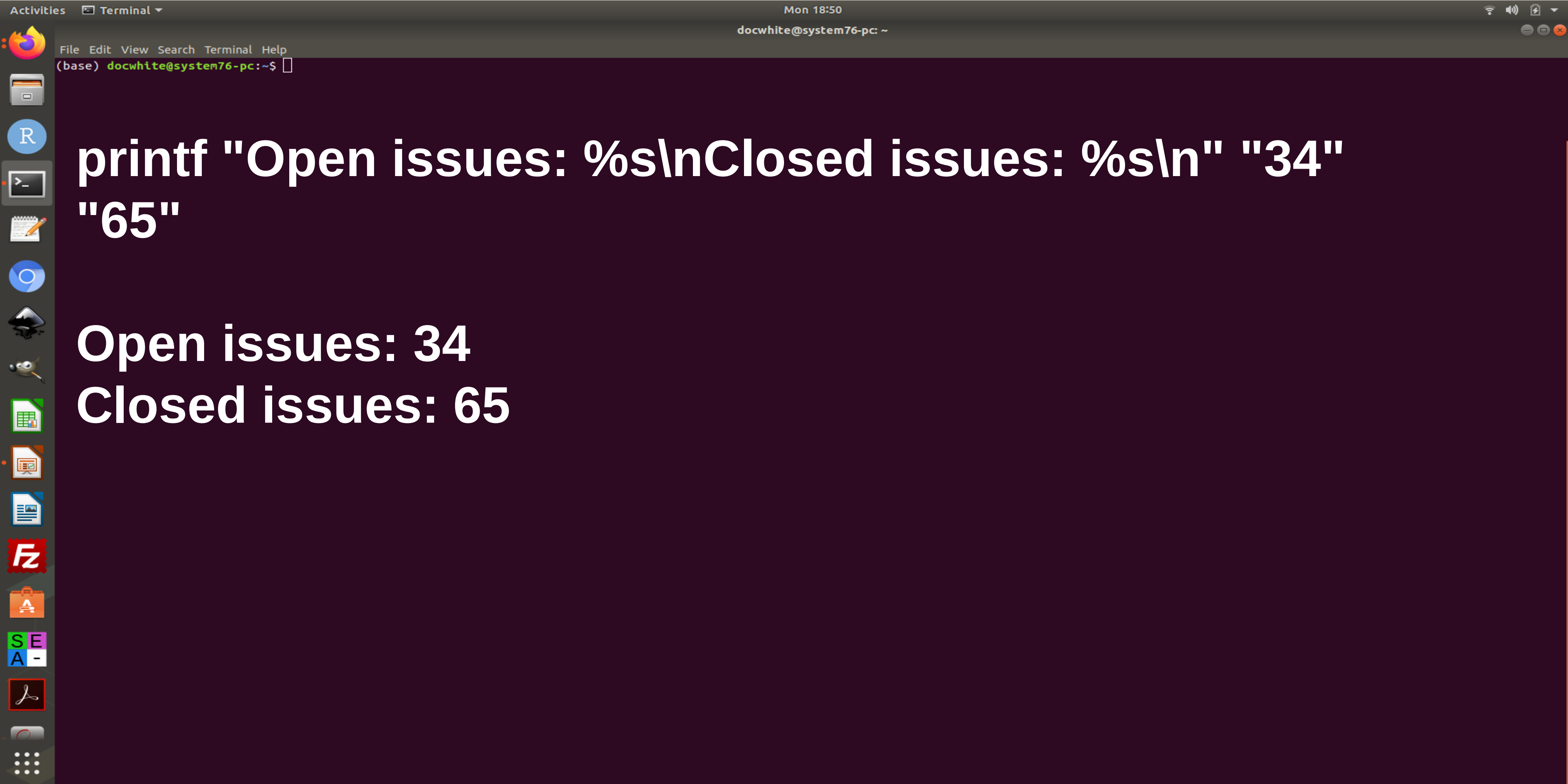
printf examples



```
(base) docwhite@system76-pc:~$
```

```
printf "Open issues: %s\nClosed issues: %s\n" "34" "65"
```

printf examples

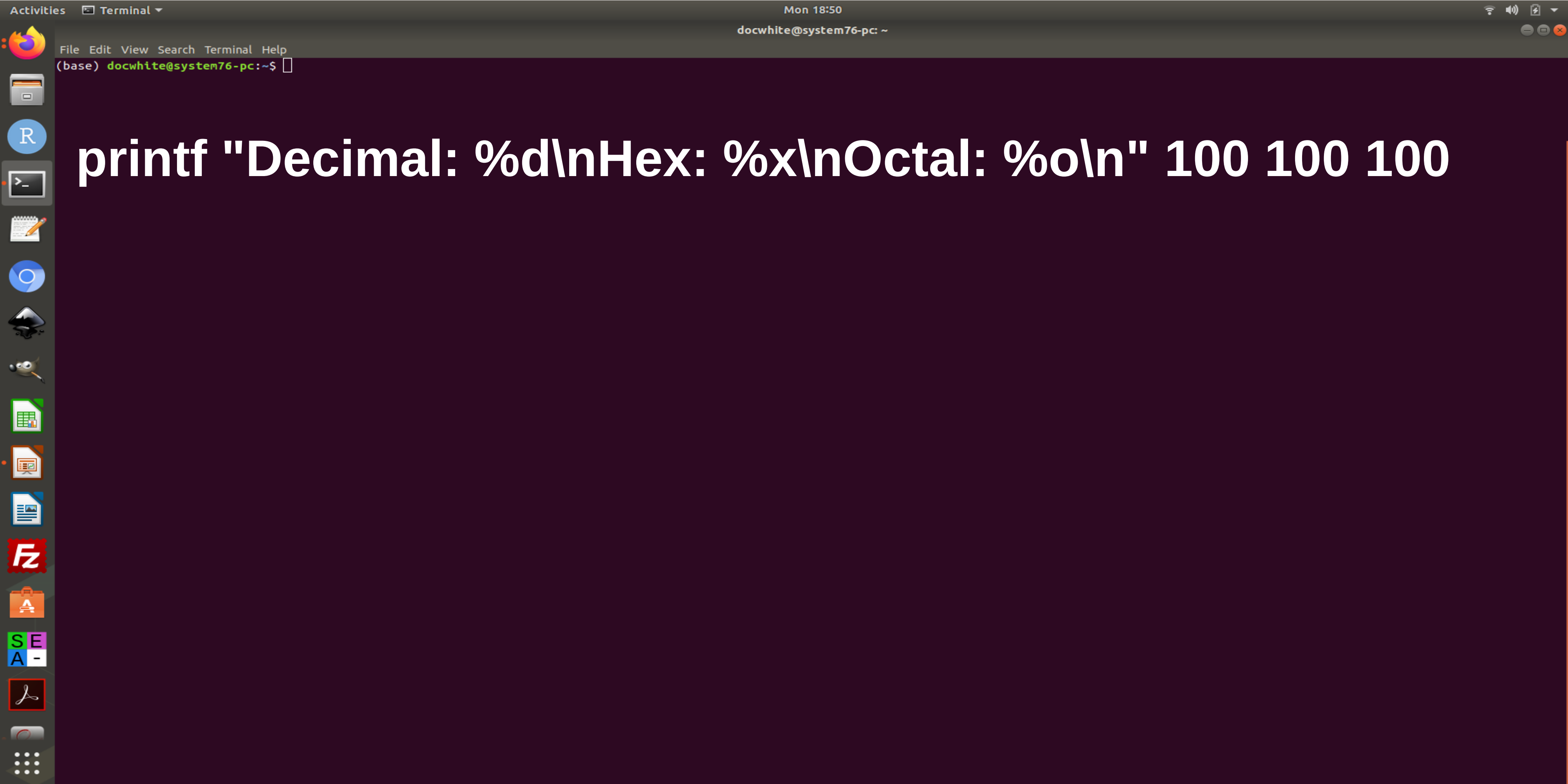


```
(base) docwhite@system76-pc:~$
```

```
printf "Open issues: %s\nClosed issues: %s\n" "34"
"65"
```

```
Open issues: 34
Closed issues: 65
```

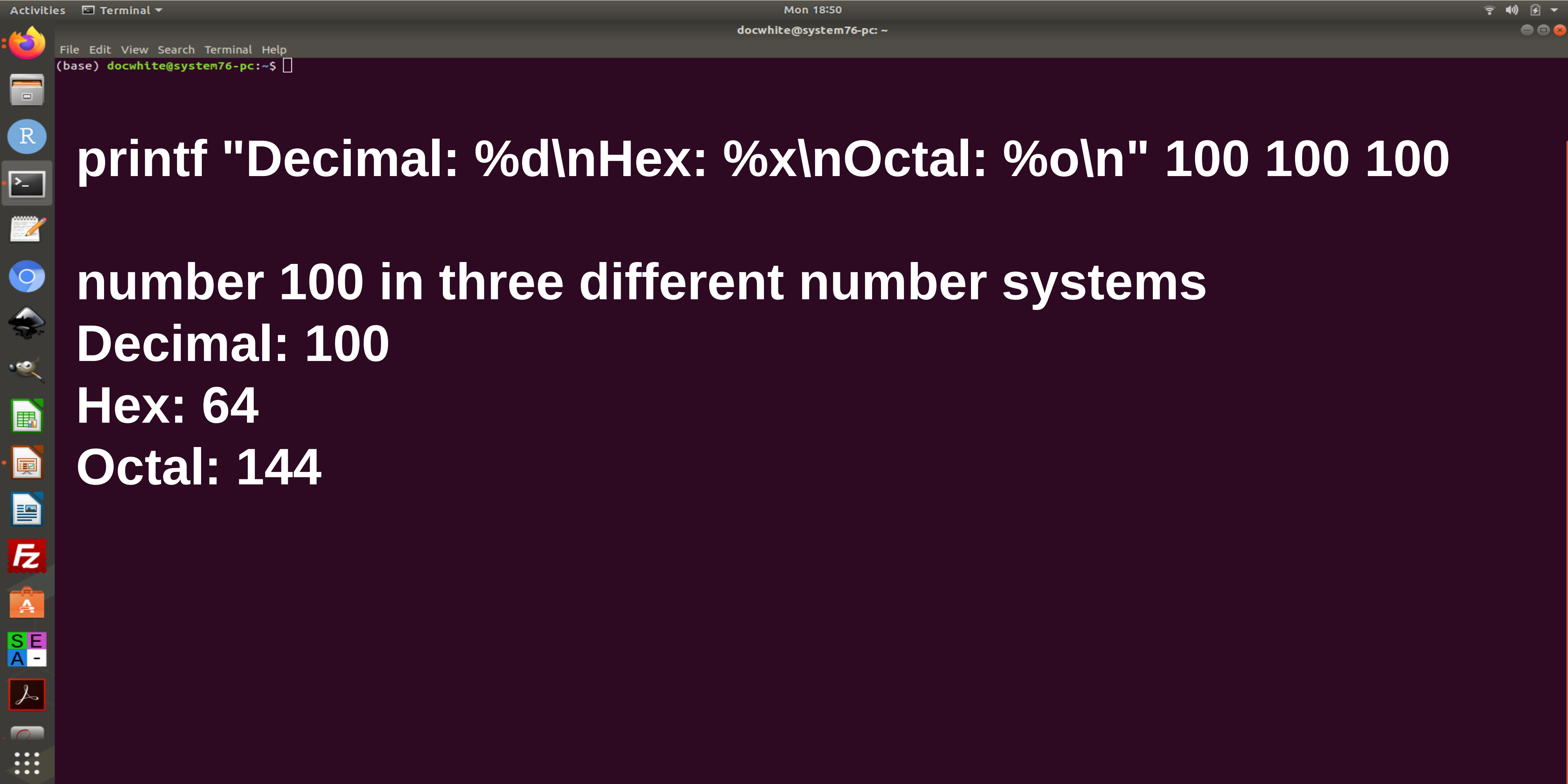
printf examples



```
(base) docwhite@system76-pc:~$
```

```
printf "Decimal: %d\nHex: %x\nOctal: %o\n" 100 100 100
```

printf examples



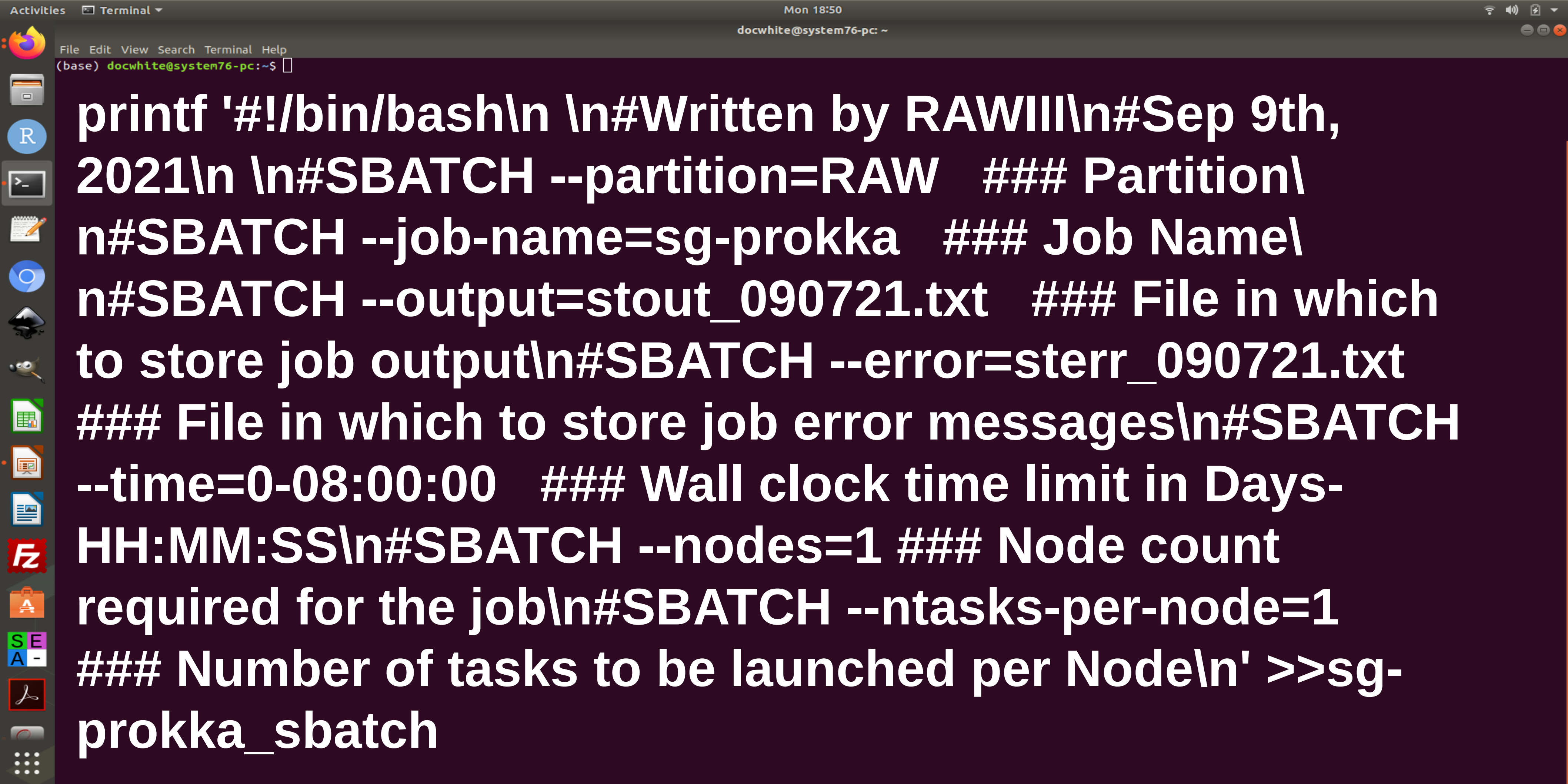
```
(base) docwhite@system76-pc:~$
```

```
printf "Decimal: %d\nHex: %x\nOctal: %o\n" 100 100 100
```

number 100 in three different number systems

```
Decimal: 100
Hex: 64
Octal: 144
```

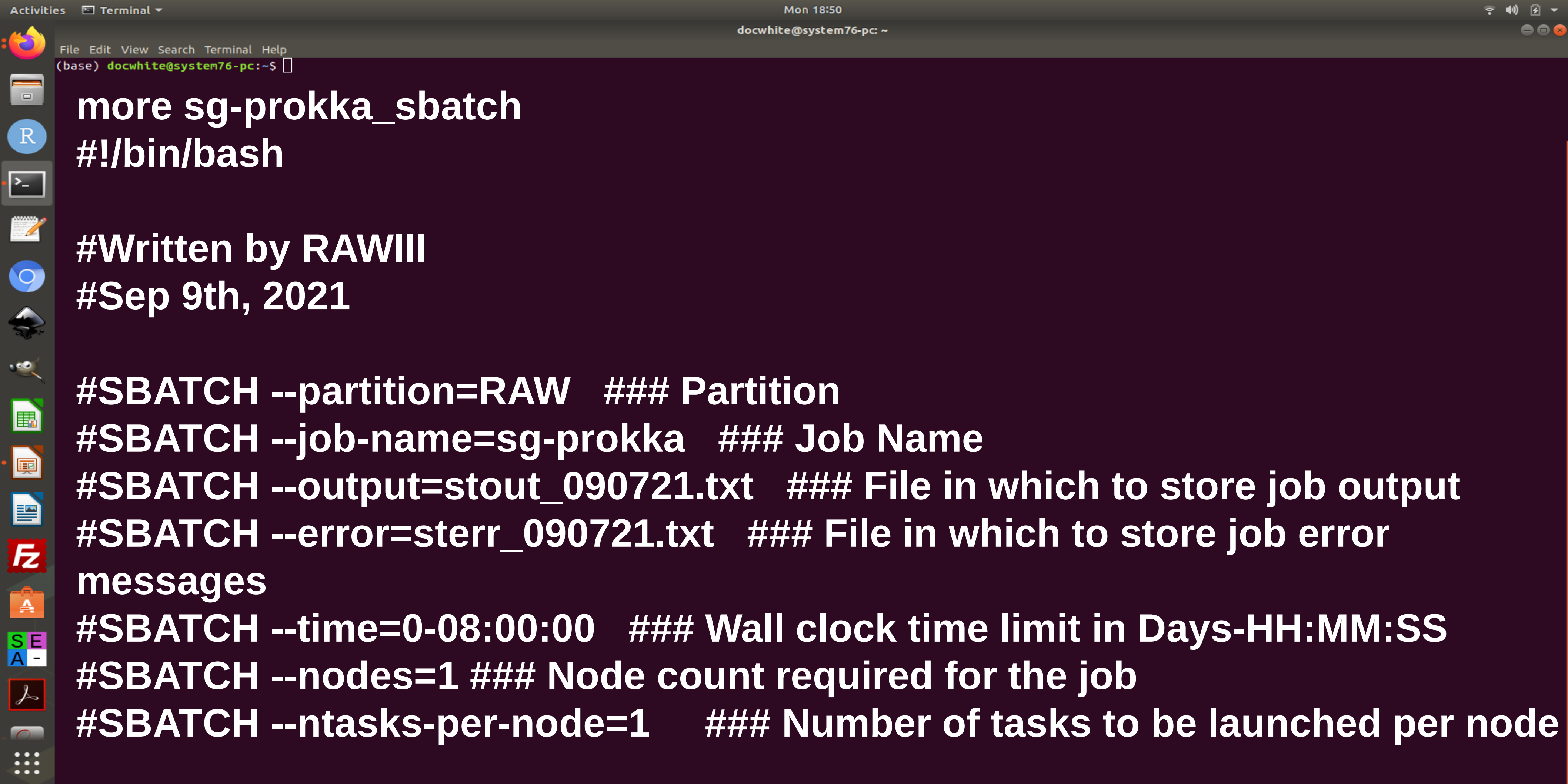
printf examples



```
(base) docwhite@system76-pc:~$
```

```
printf '#!/bin/bash\n \n#Written by RAWIII\n#Sep 9th, 2021\n \n#SBATCH --partition=RAW   ### Partition\n#SBATCH --job-name=sg-prokka   ### Job Name\n#SBATCH --output=stout_090721.txt   ### File in which to store job output\n#SBATCH --error=sterr_090721.txt ### File in which to store job error messages\n#SBATCH --time=0-08:00:00   ### Wall clock time limit in Days-HH:MM:SS\n#SBATCH --nodes=1 ### Node count required for the job\n#SBATCH --ntasks-per-node=1 ### Number of tasks to be launched per Node\n' >>sg-prokka_sbatch
```

printf examples



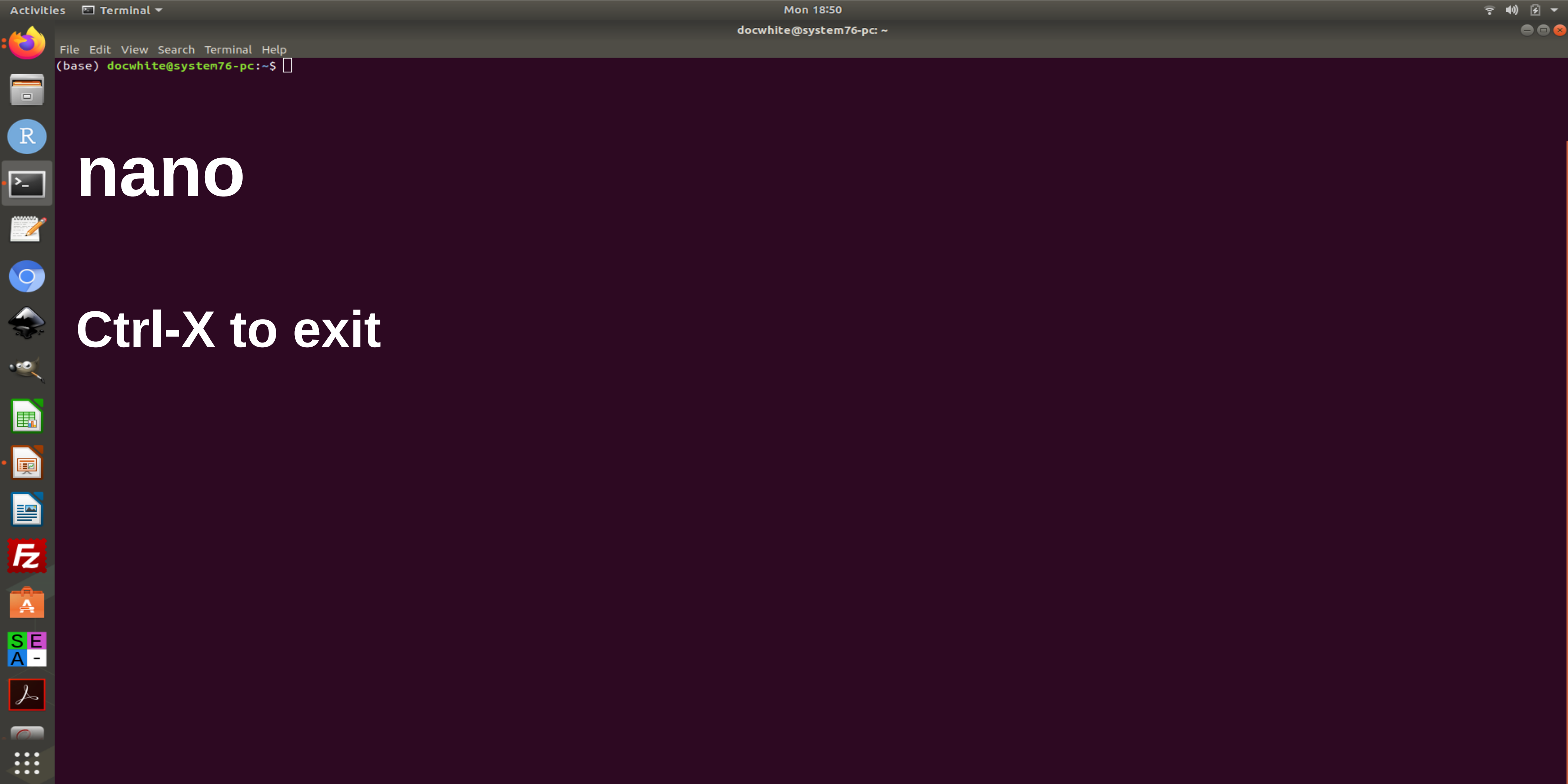
```
(base) docwhite@system76-pc:~$
```

```
more sg-prokka_sbatch
#!/bin/bash

#Written by RAWIII
#Sep 9th, 2021

#SBATCH --partition=RAW   ### Partition
#SBATCH --job-name=sg-prokka   ### Job Name
#SBATCH --output=stout_090721.txt   ### File in which to store job output
#SBATCH --error=sterr_090721.txt   ### File in which to store job error
messages
#SBATCH --time=0-08:00:00   ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1 ### Node count required for the job
#SBATCH --ntasks-per-node=1      ### Number of tasks to be launched per node
```

Text editors - nano

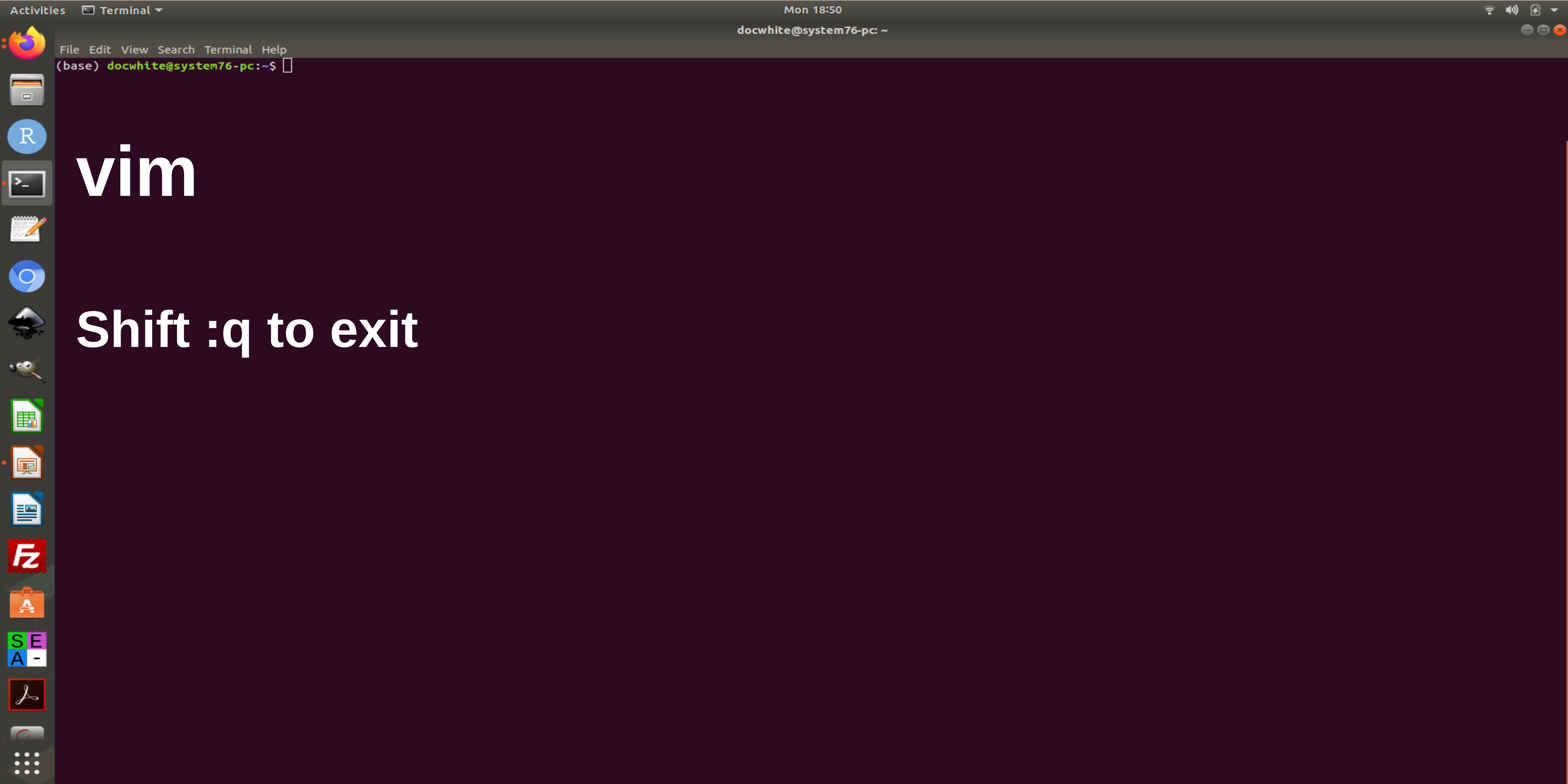


```
(base) docwhite@system76-pc:~$ 
```

nano

Ctrl-X to exit

Text editors - vim

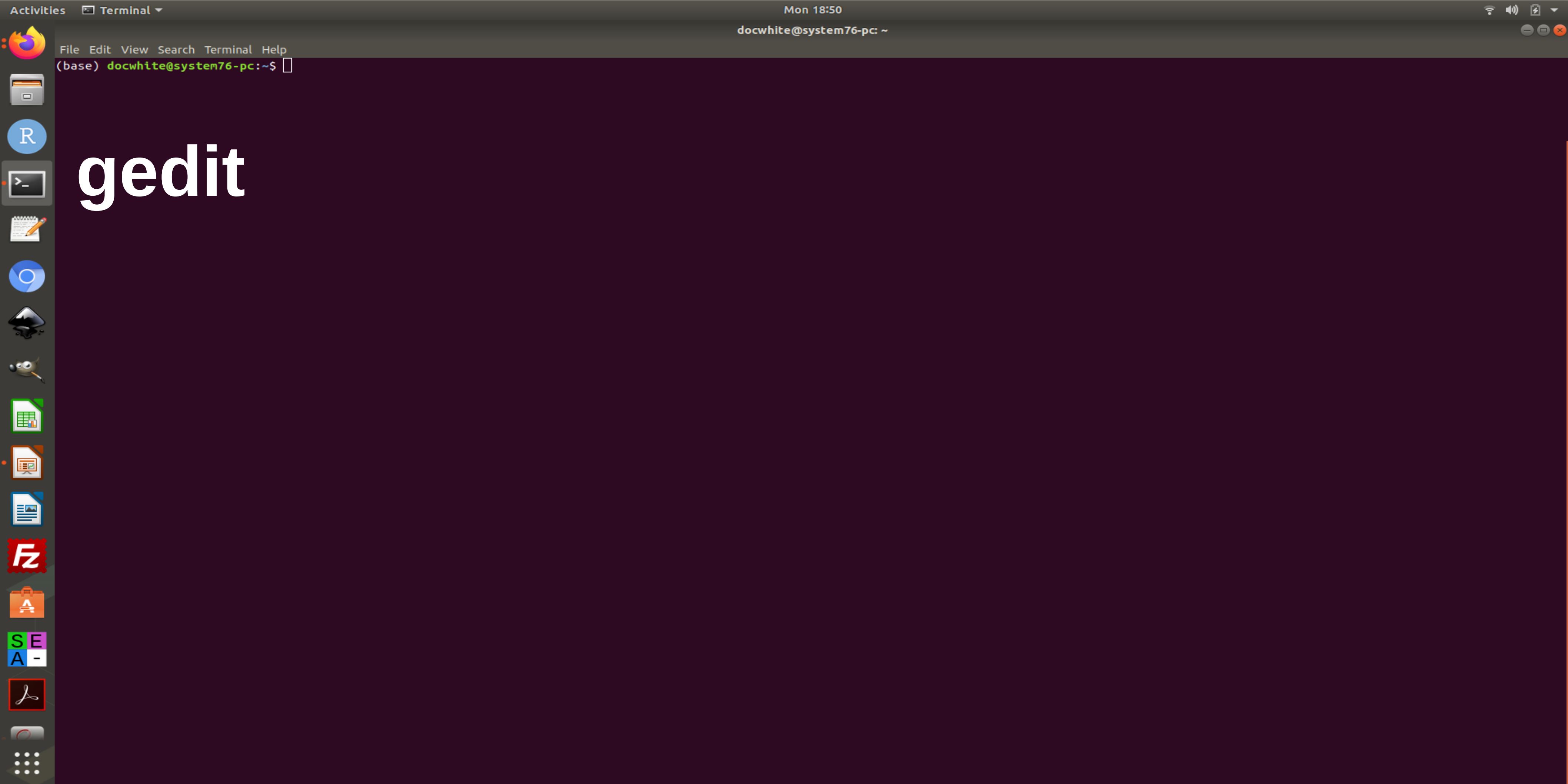


```
(base) docwhite@system76-pc:~$ 
```

vim

Shift :q to exit

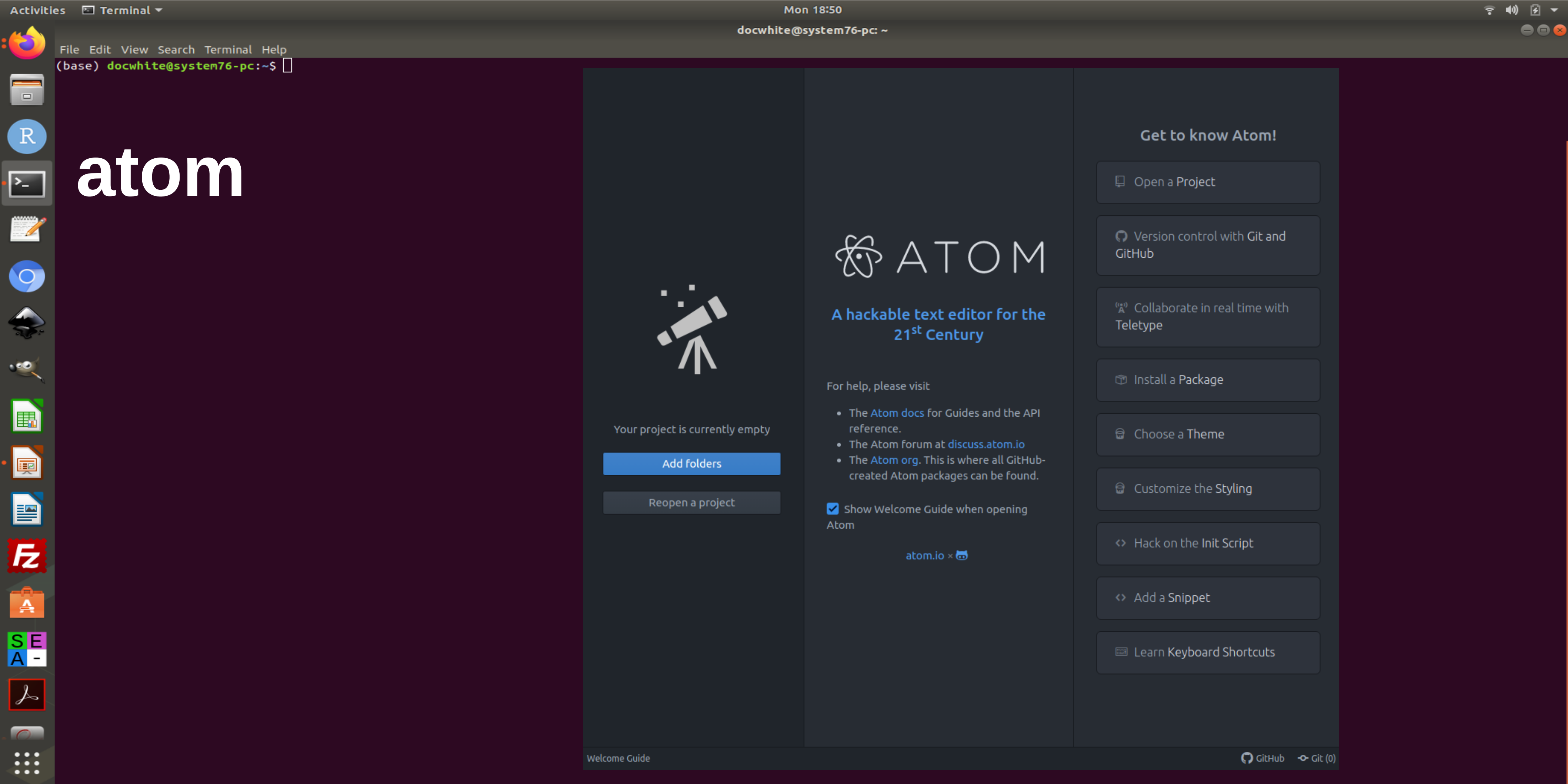
Text editors - gedit



```
(base) docwhite@system76-pc:~$ 
```

gedit

Text editors - atom



```
(base) docwhite@system76-pc:~$ 
```

atom

A hackable text editor for the 21^{st} Century

For help, please visit

- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io
- The Atom org. This is where all GitHub-created Atom packages can be found.

☑ Show Welcome Guide when opening Atom

atom.io ×

Get to know Atom!

cut – syntax anatomy UNIX’s scissors

cut [options] file.txt

-d (--delimiter) “,” set field delimiter (default tab)

-f (--fields=LIST) Select by specifying a field

- -f 2 select a field to cut (left is 1)
- -f 2-8,12 select multiple fields to cut

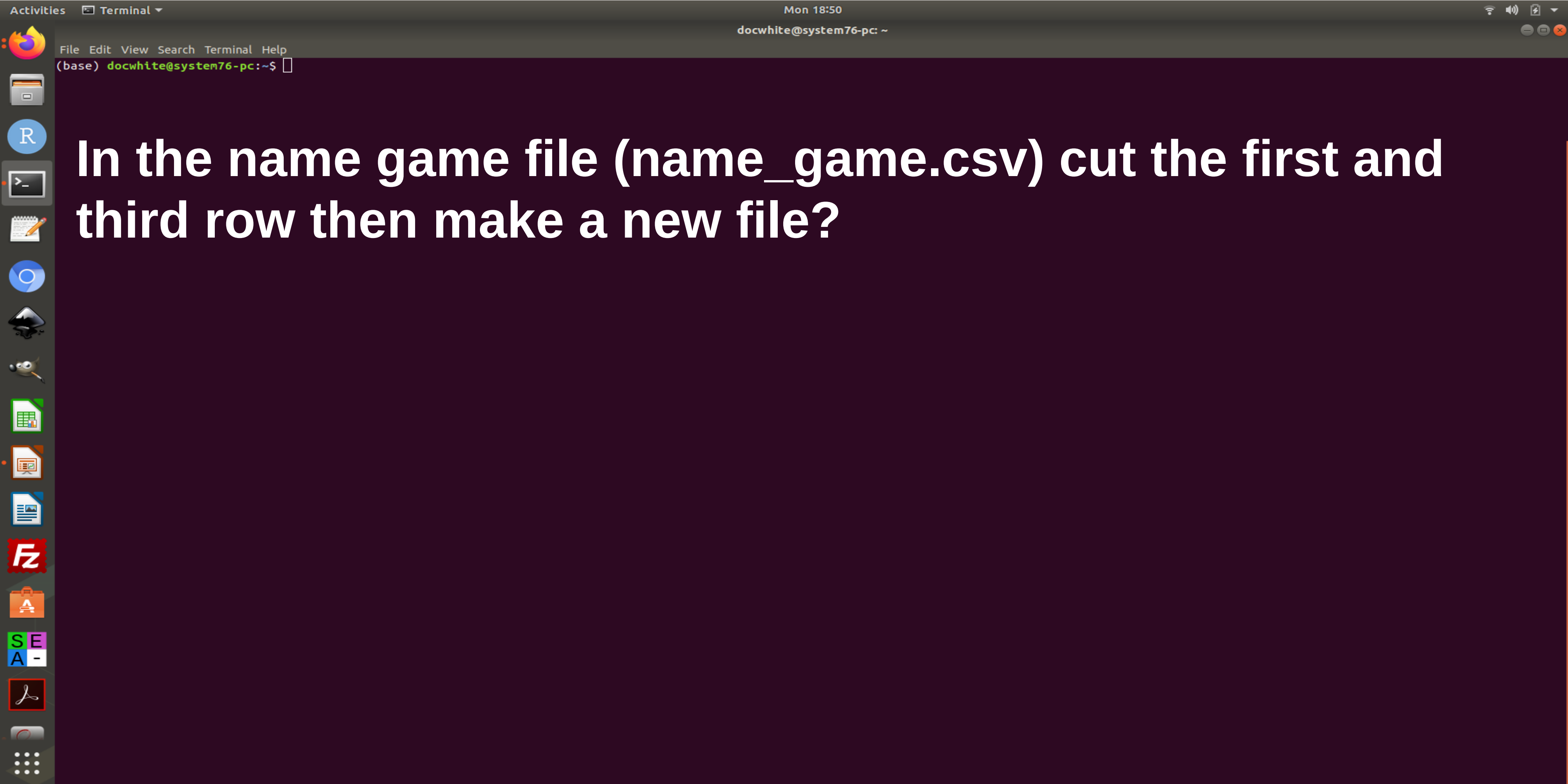
-b (--bytes=LIST) Select by specifying a byte

-c (--characters=LIST) Select by specifying a character

--complement - Complement the selection.

-s (--only-delimited) suppress non-matches

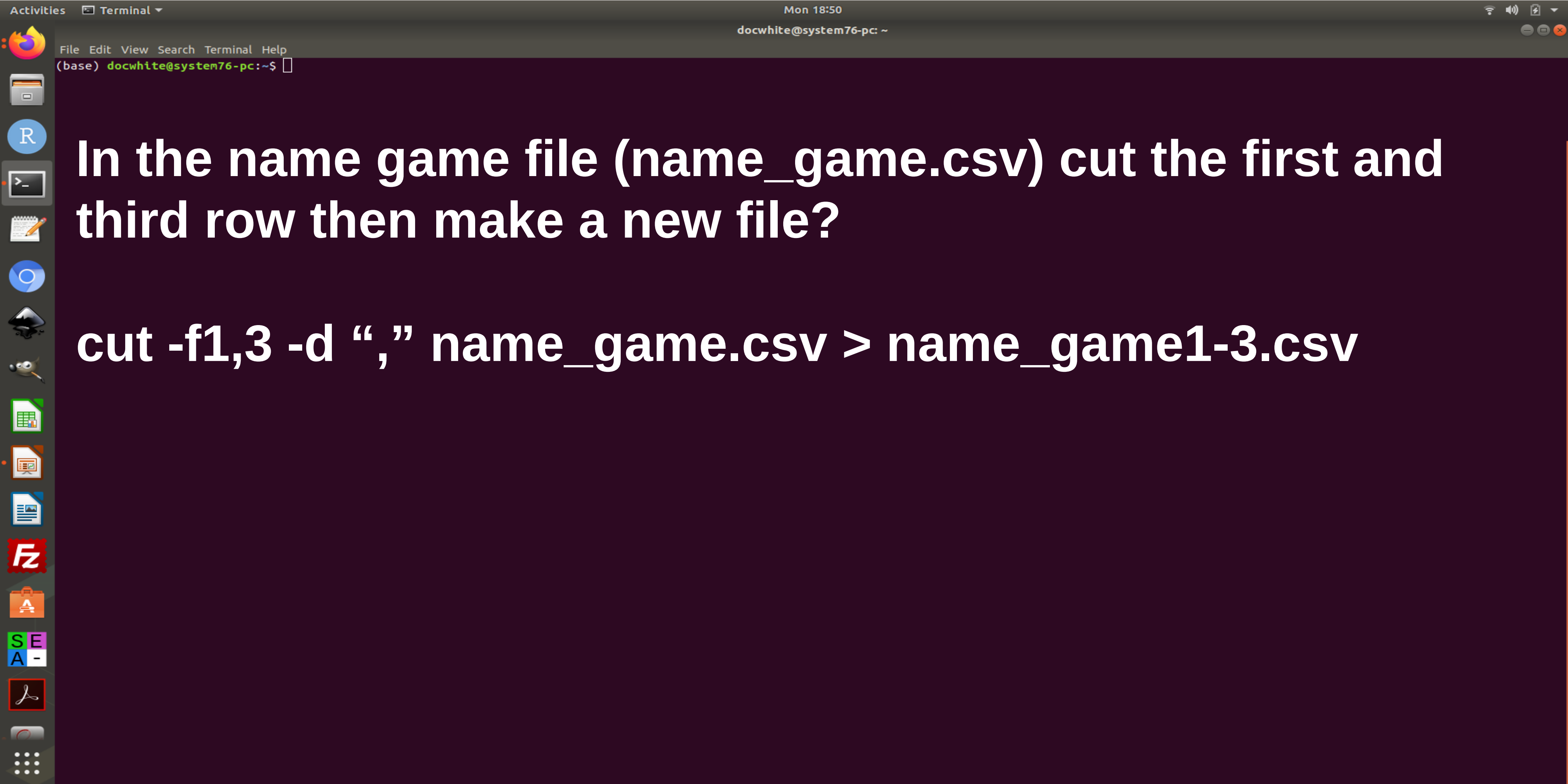
Cut examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.csv) cut the first and third row then make a new file?

Cut examples

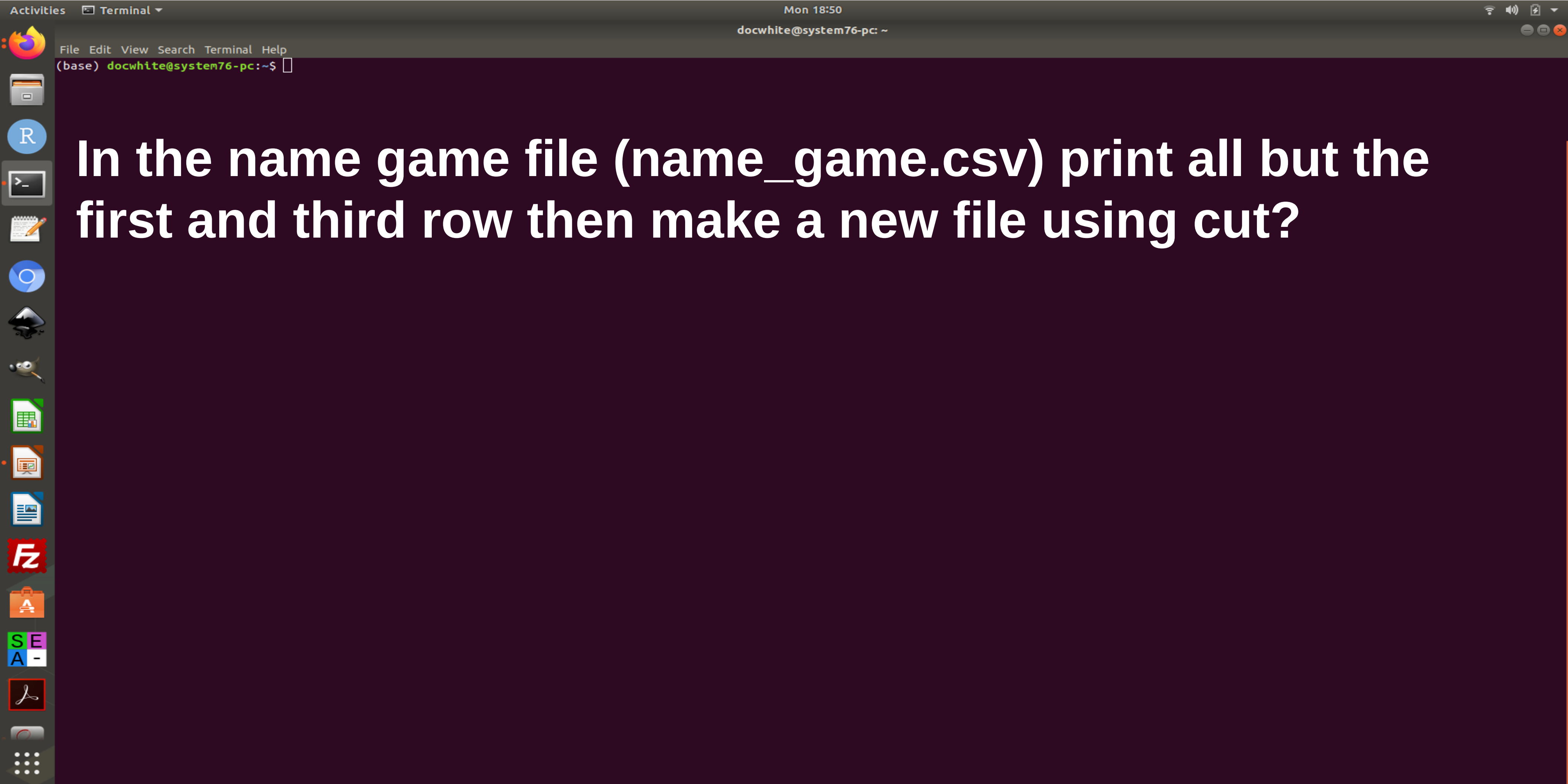


```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.csv) cut the first and third row then make a new file?

```
cut -f1,3 -d “,” name_game.csv > name_game1-3.csv
```

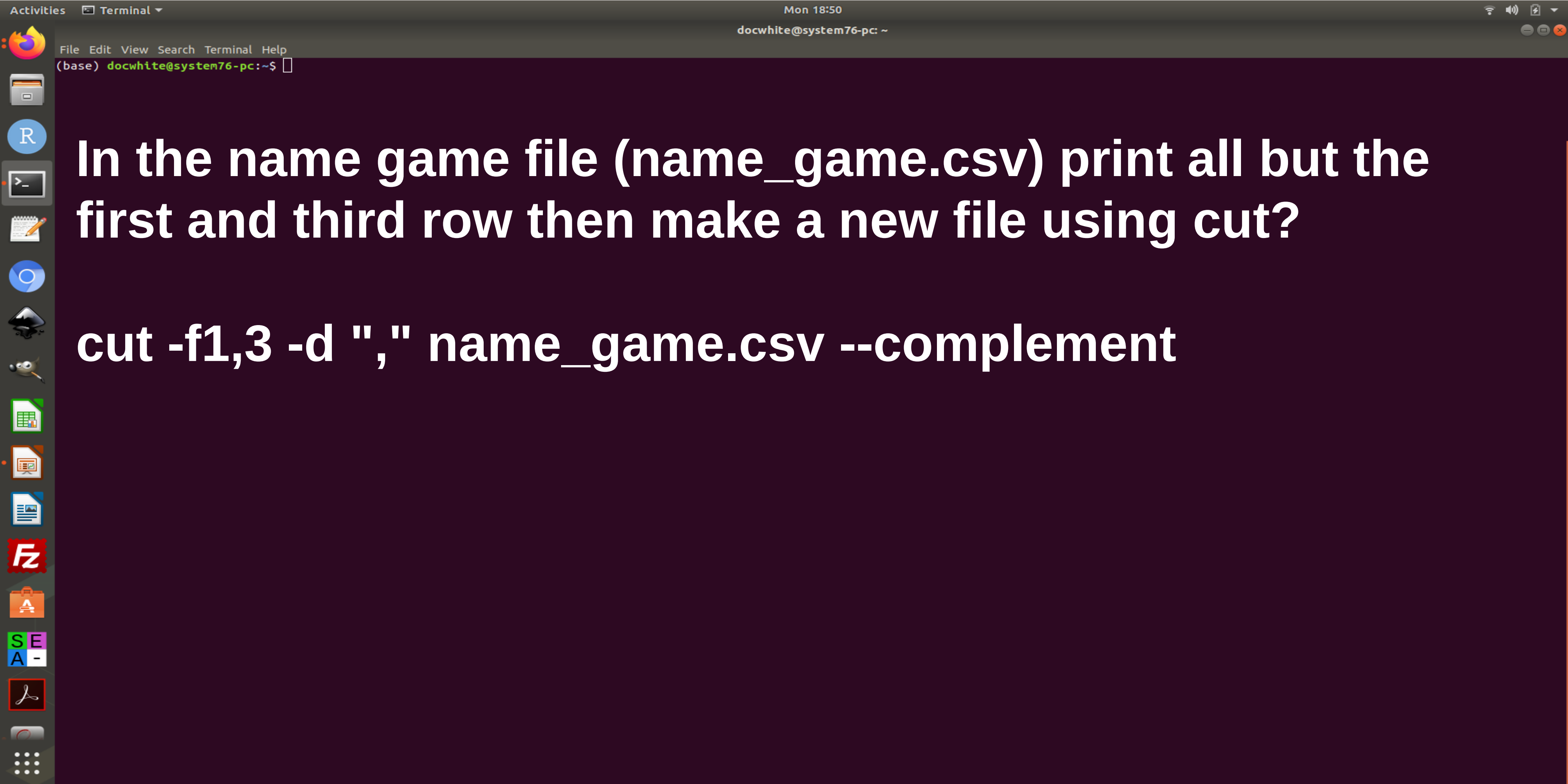
Cut examples



```
(base) docwhite@system76-pc:~$
```

In the name game file (name_game.csv) print all but the first and third row then make a new file using cut?

Cut examples



```
(base) docwhite@system76-pc:~$
```

In the name game file (name_game.csv) print all but the first and third row then make a new file using cut?

```
cut -f1,3 -d "," name_game.csv --complement
```

sort – syntax anatomy of sort

sort [options] file.txt

-t “t” set the delimiter when using -k

default is non-blank to blank transition

-k 3   sort column #3 (left is 1)

-k 2,3 sort multiple columns

-n sort numerically

-r reverse sort order

-u drop duplicates from the result

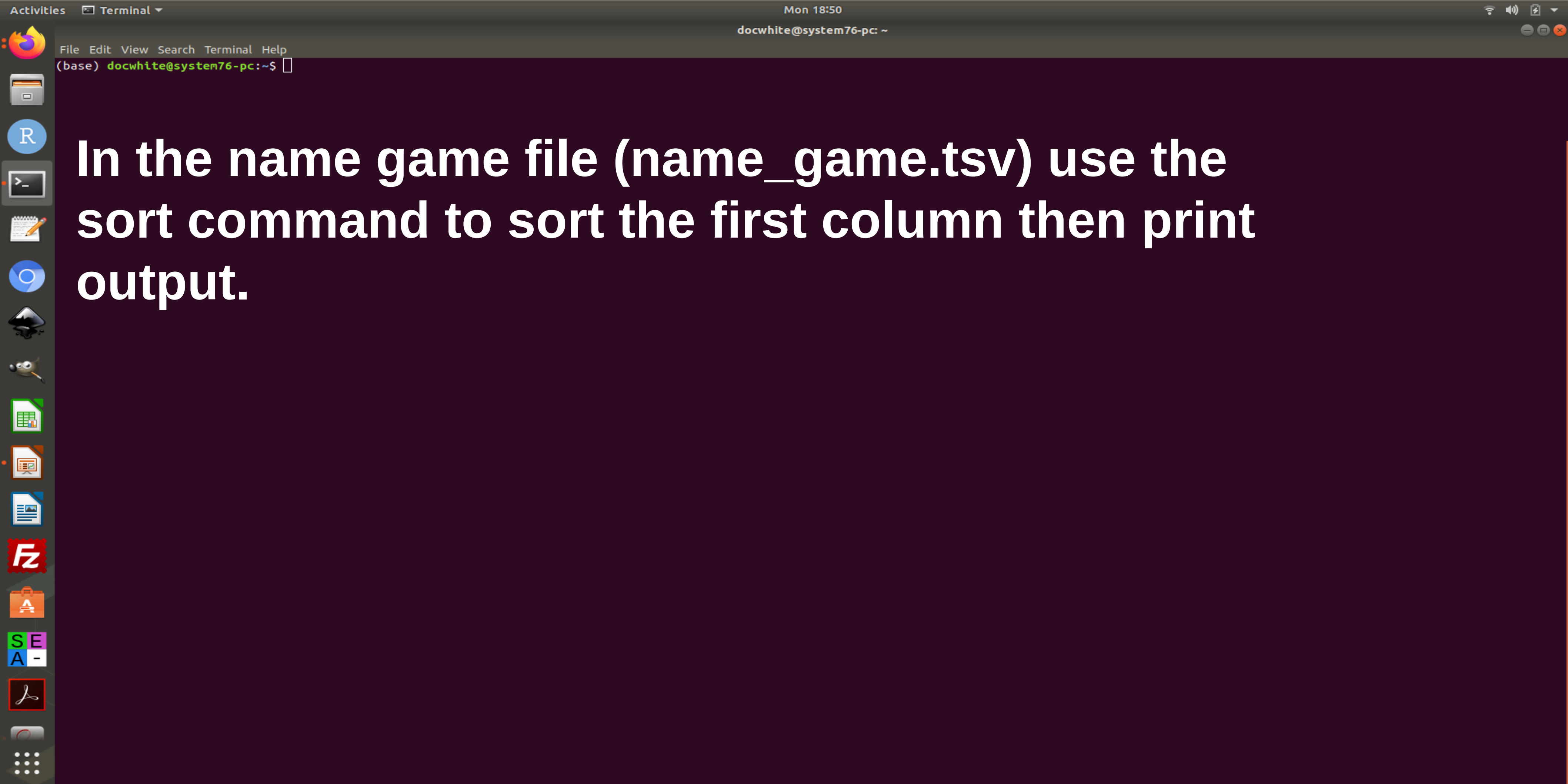
-b, --ignore-leading-blanks, ignore leading blanks

-d, --dictionary-order consider only blanks and alphanumeric characters

-f, --ignore-case fold lower case to upper case characters

No options: sort alphabetically from leftmost character.

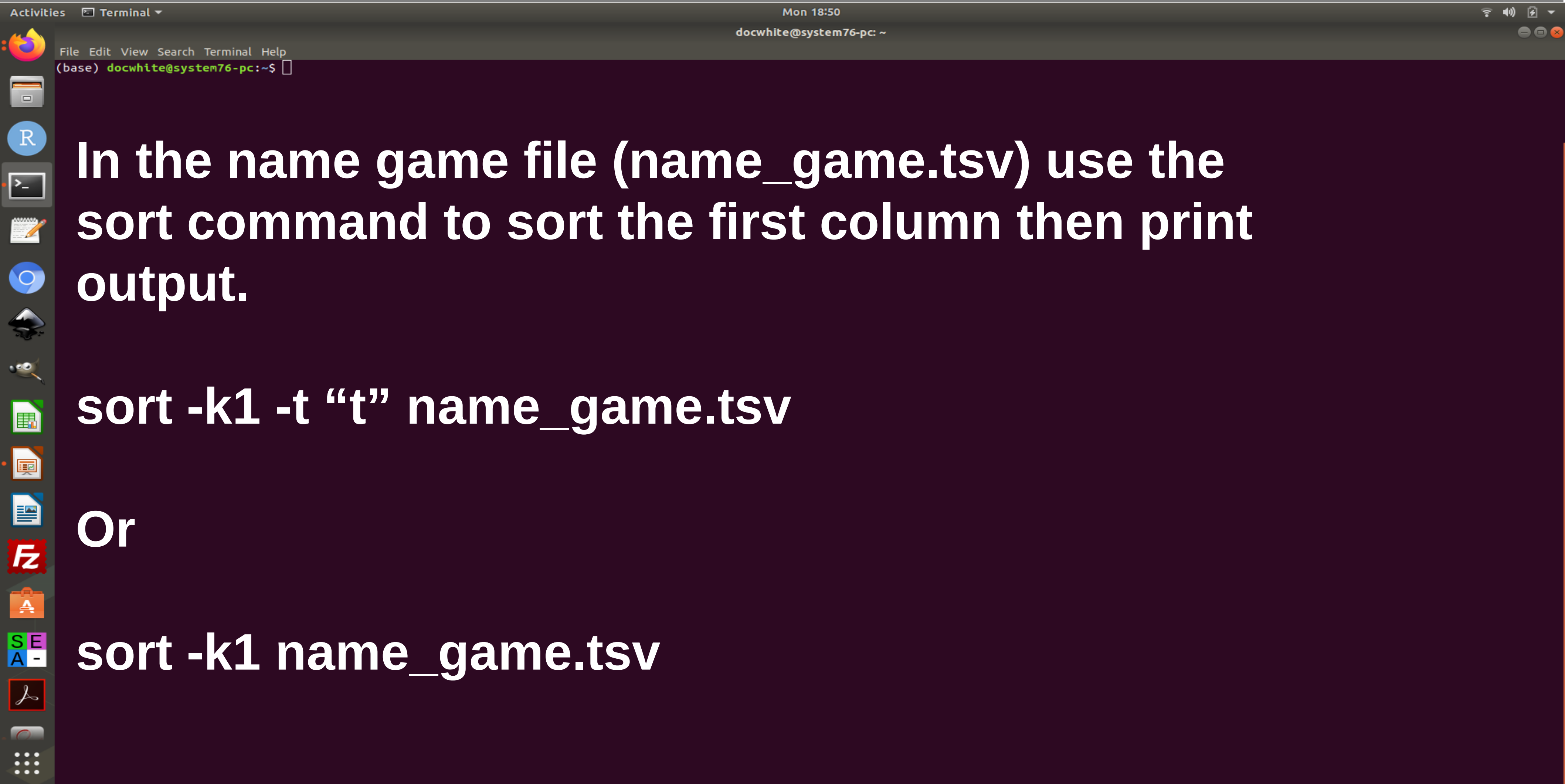
Sort examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.tsv) use the sort command to sort the first column then print output.

Sort examples



```
(base) docwhite@system76-pc:~$ 
```

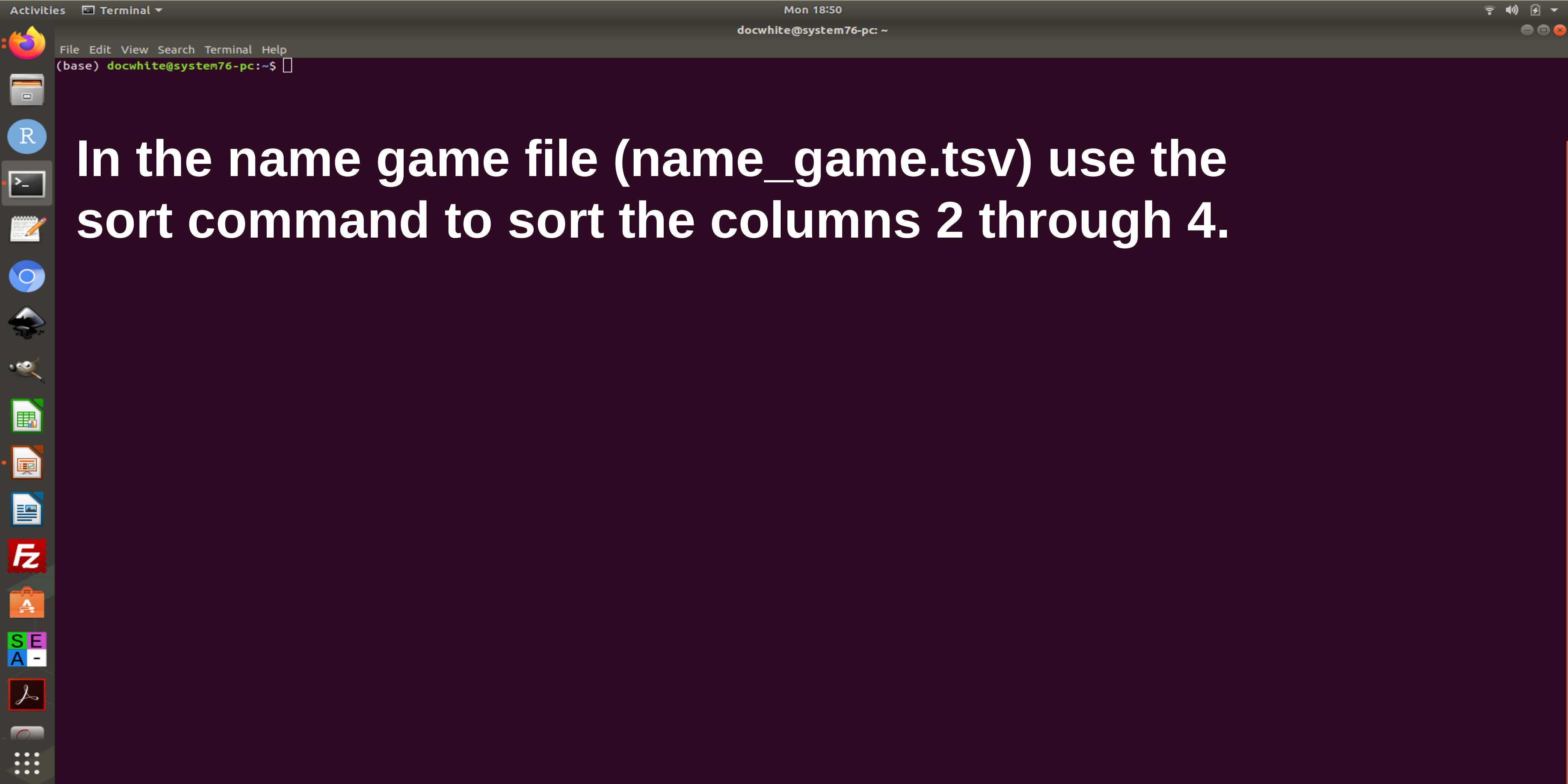
In the name game file (name_game.tsv) use the sort command to sort the first column then print output.

sort -k1 -t “t” name_game.tsv

Or

sort -k1 name_game.tsv

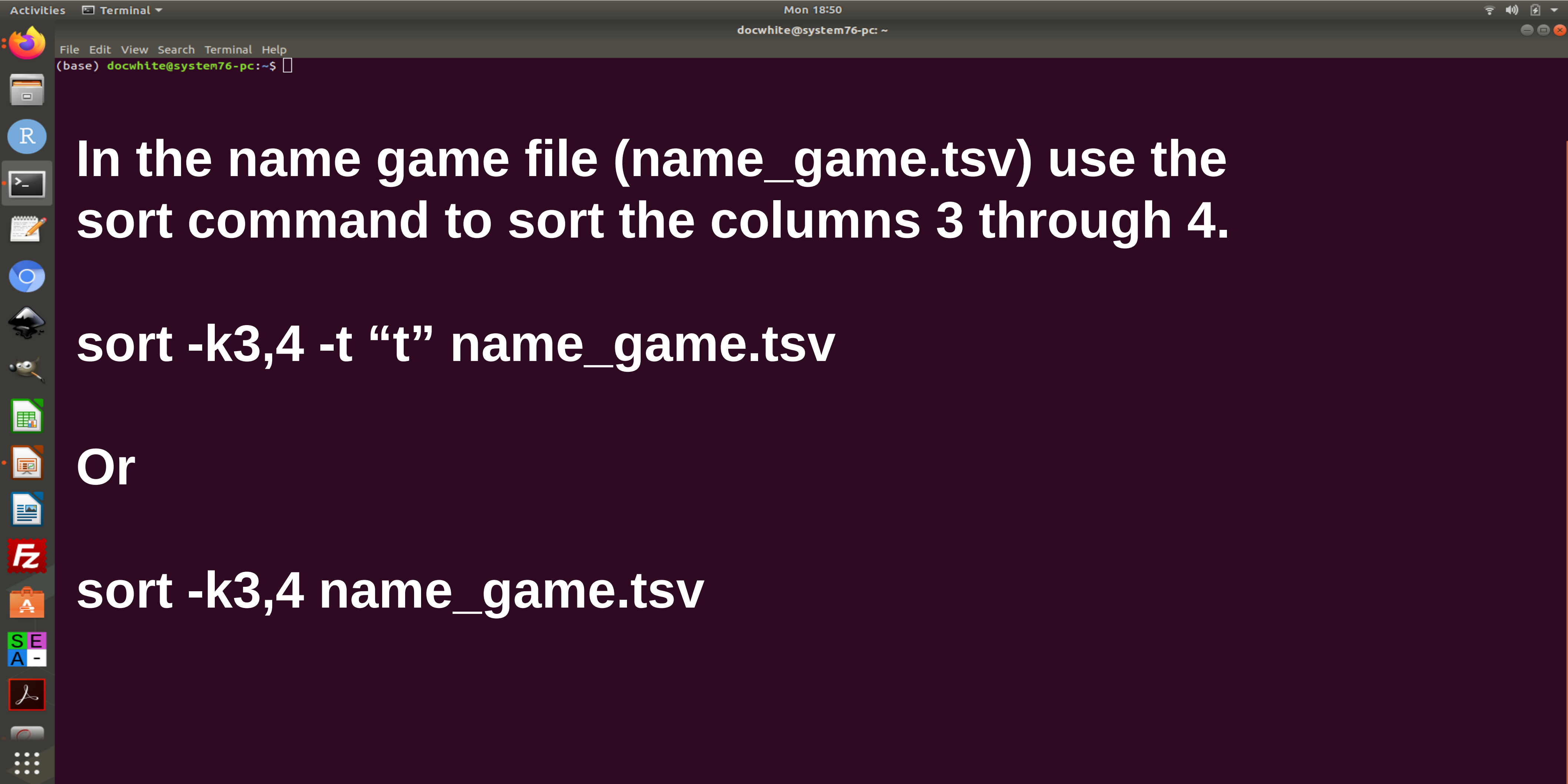
Sort examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.tsv) use the sort command to sort the columns 2 through 4.

Sort examples



```
(base) docwhite@system76-pc:~$
```

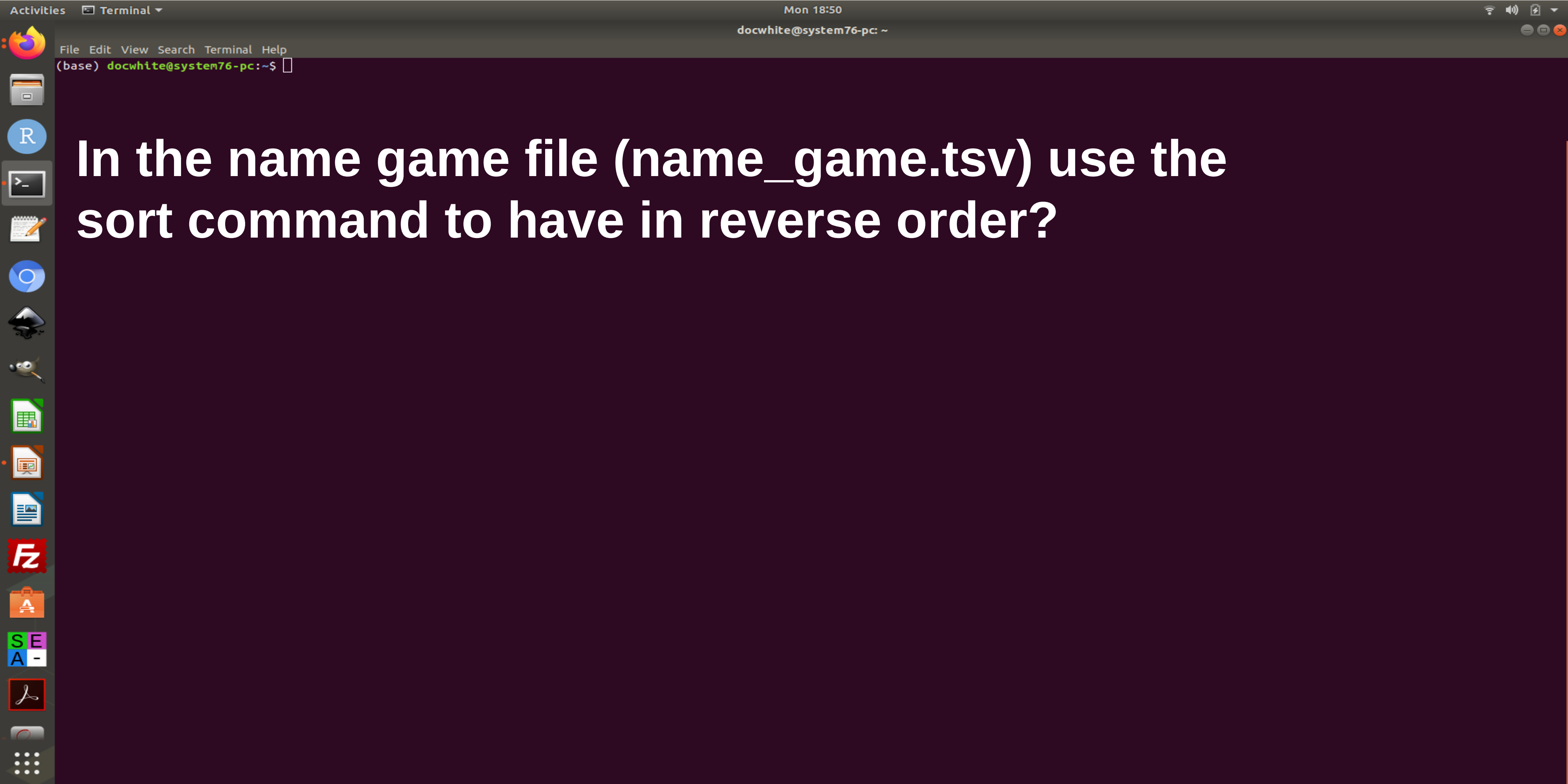
In the name game file (name_game.tsv) use the sort command to sort the columns 3 through 4.

```
sort -k3,4 -t “t” name_game.tsv
```

Or

```
sort -k3,4 name_game.tsv
```

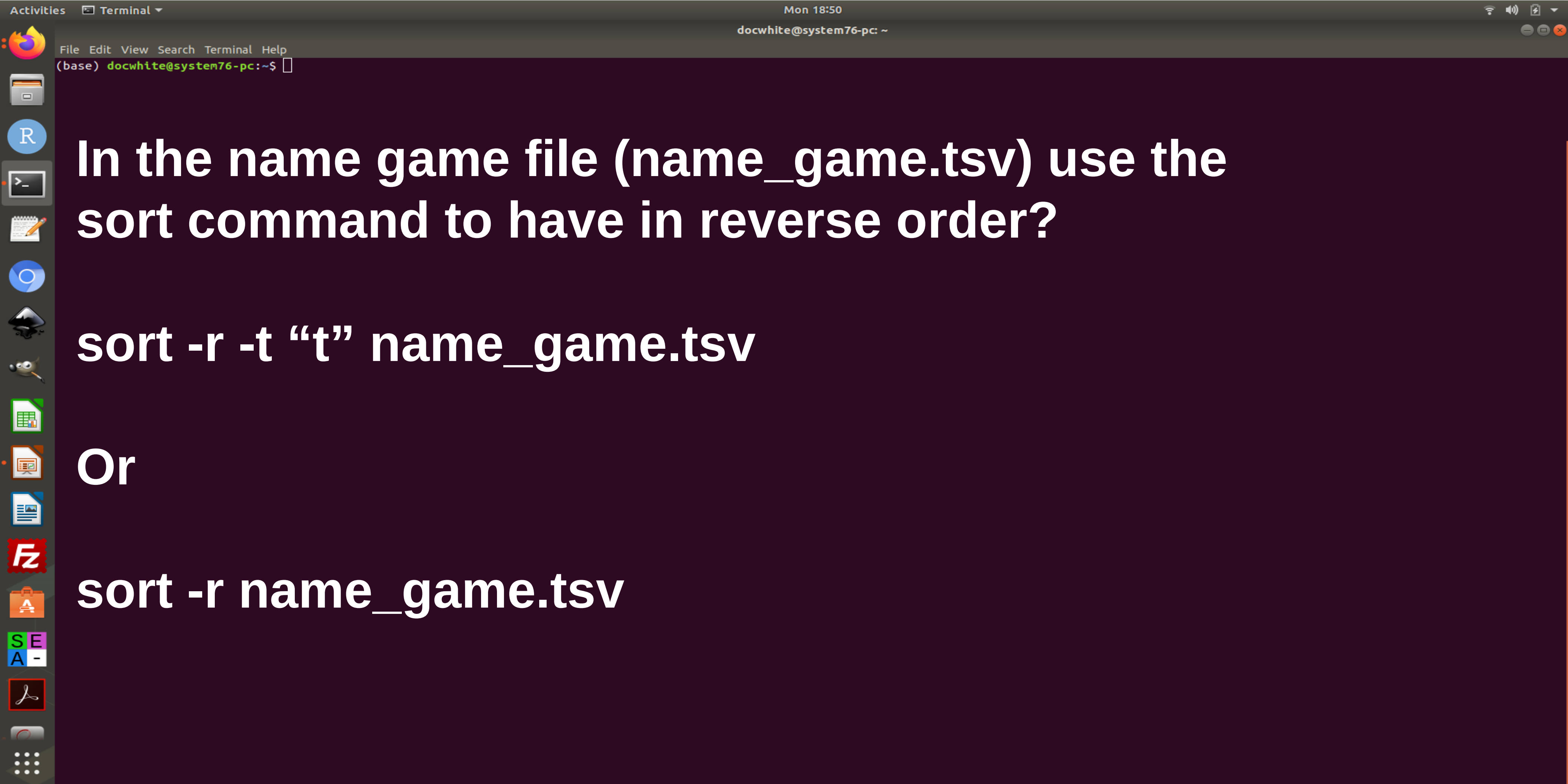
Sort examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.tsv) use the sort command to have in reverse order?

Sort examples



```
(base) docwhite@system76-pc:~$ 
```

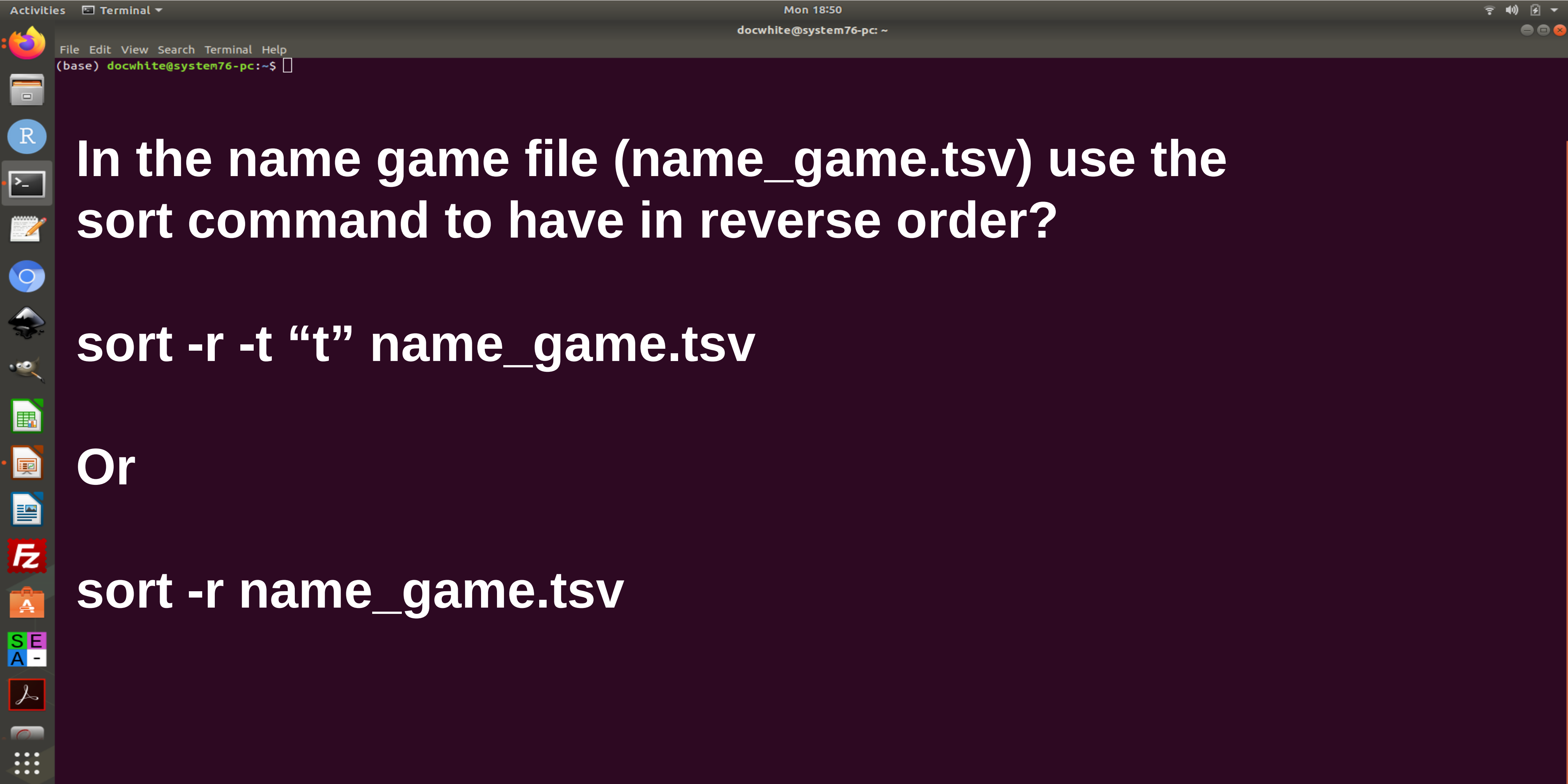
In the name game file (name_game.tsv) use the sort command to have in reverse order?

sort -r -t “t” name_game.tsv

Or

sort -r name_game.tsv

Sort examples



```
(base) docwhite@system76-pc:~$ 
```

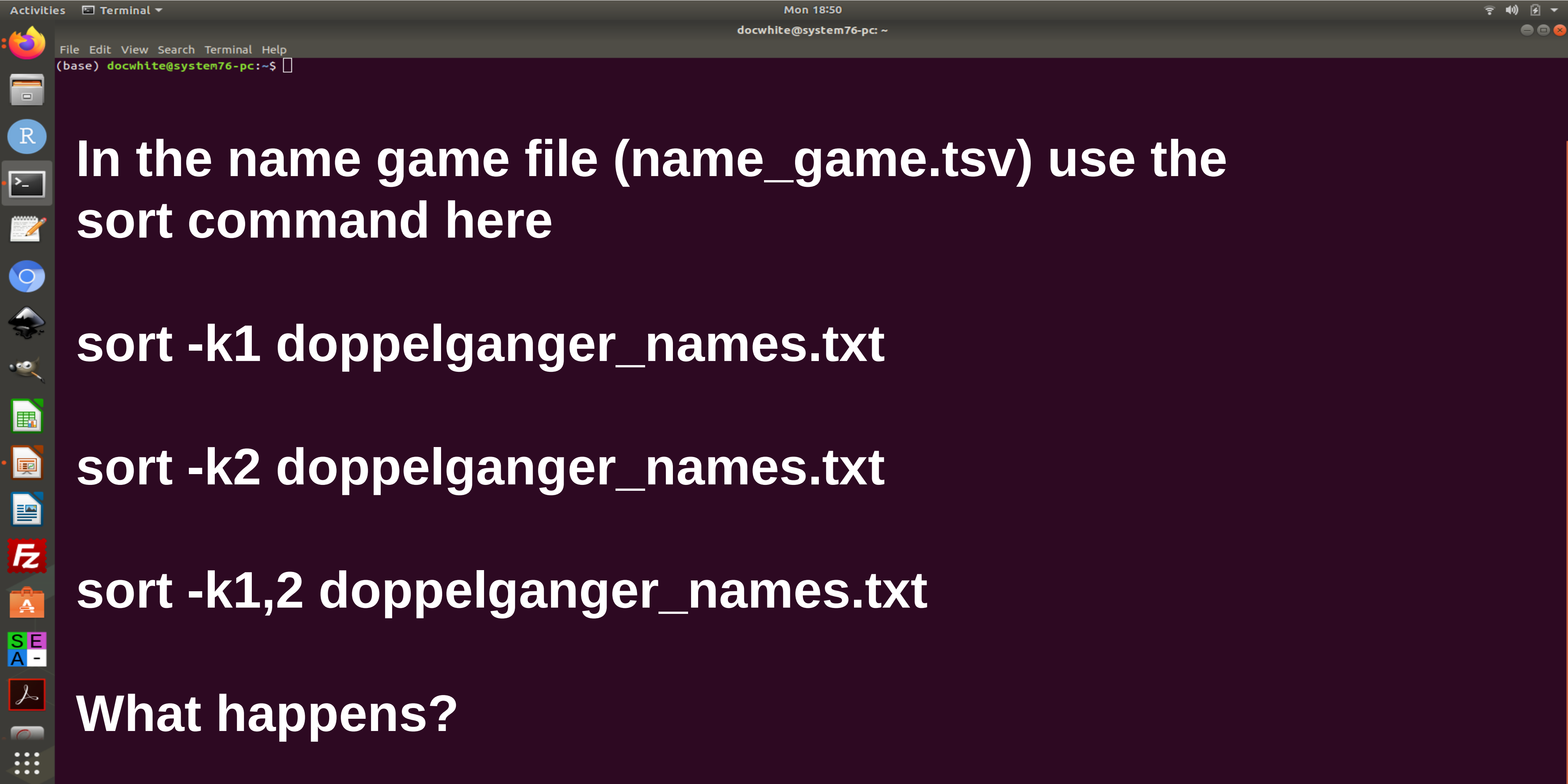
In the name game file (name_game.tsv) use the sort command to have in reverse order?

sort -r -t “t” name_game.tsv

Or

sort -r name_game.tsv

Sort examples



```
(base) docwhite@system76-pc:~$ 
```

In the name game file (name_game.tsv) use the sort command here

sort -k1 doppelganger_names.txt

sort -k2 doppelganger_names.txt

sort -k1,2 doppelganger_names.txt

What happens?

uniq – syntax anatomy of uniq

uniq [options] file.txt

Only works on sorted files and adjacent lines!

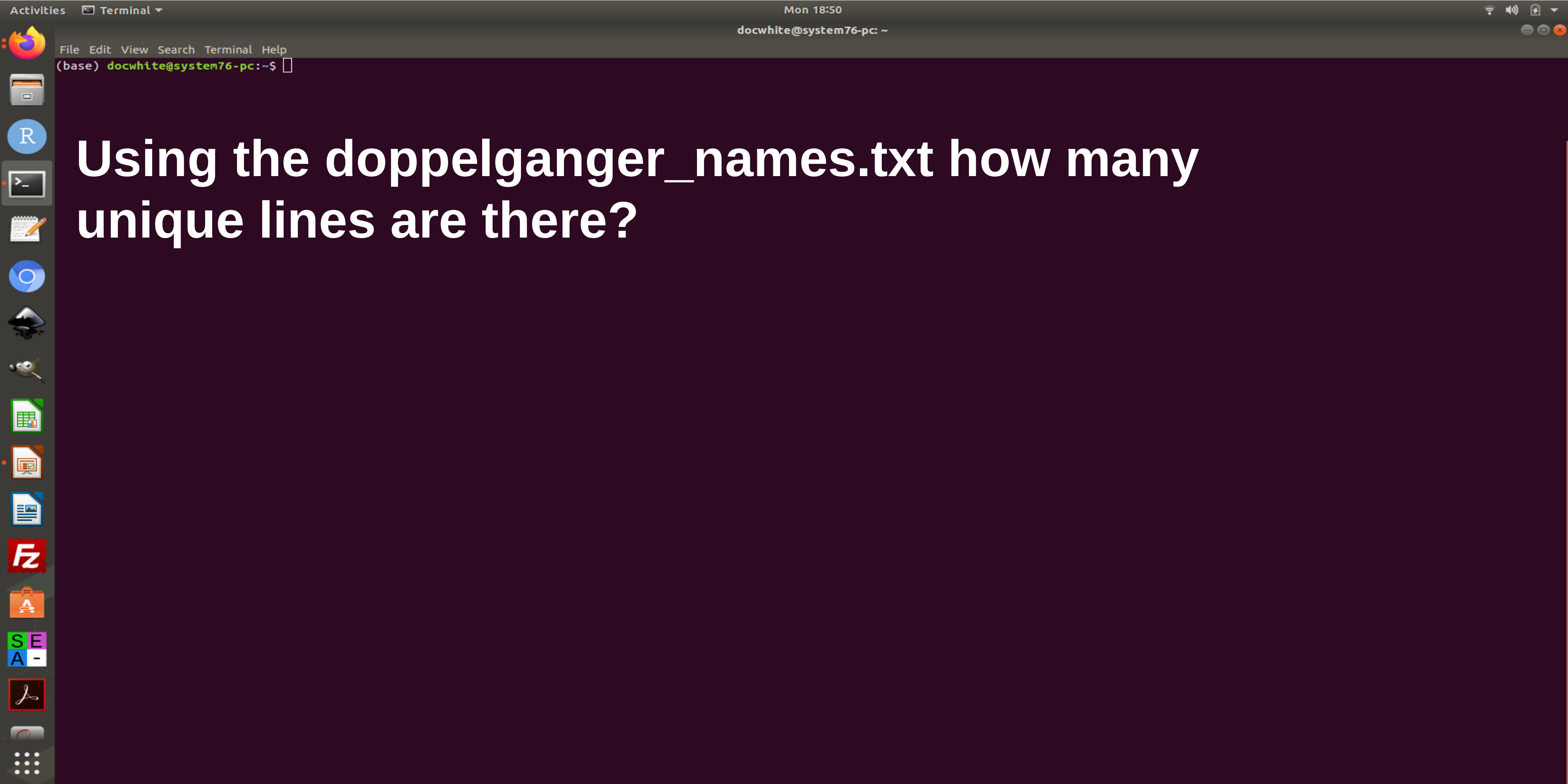
-c count lines for each unique value

-d only report duplicated lines

-u only report non-duplicated lines

No options: drop all duplicated lines (keeps 1 copy)

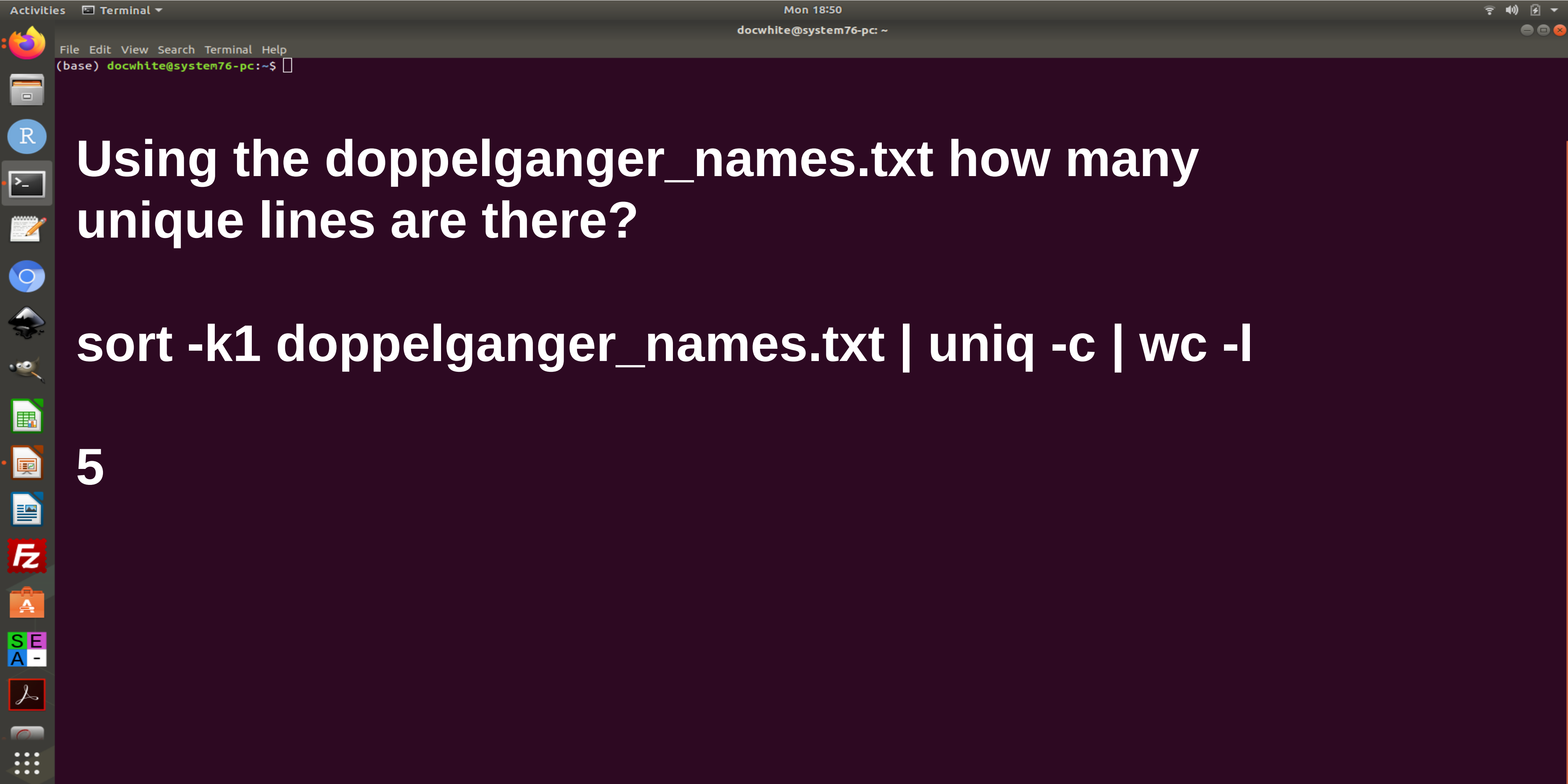
uniq examples



```
(base) docwhite@system76-pc:~$ 
```

Using the doppelganger_names.txt how many unique lines are there?

uniq examples



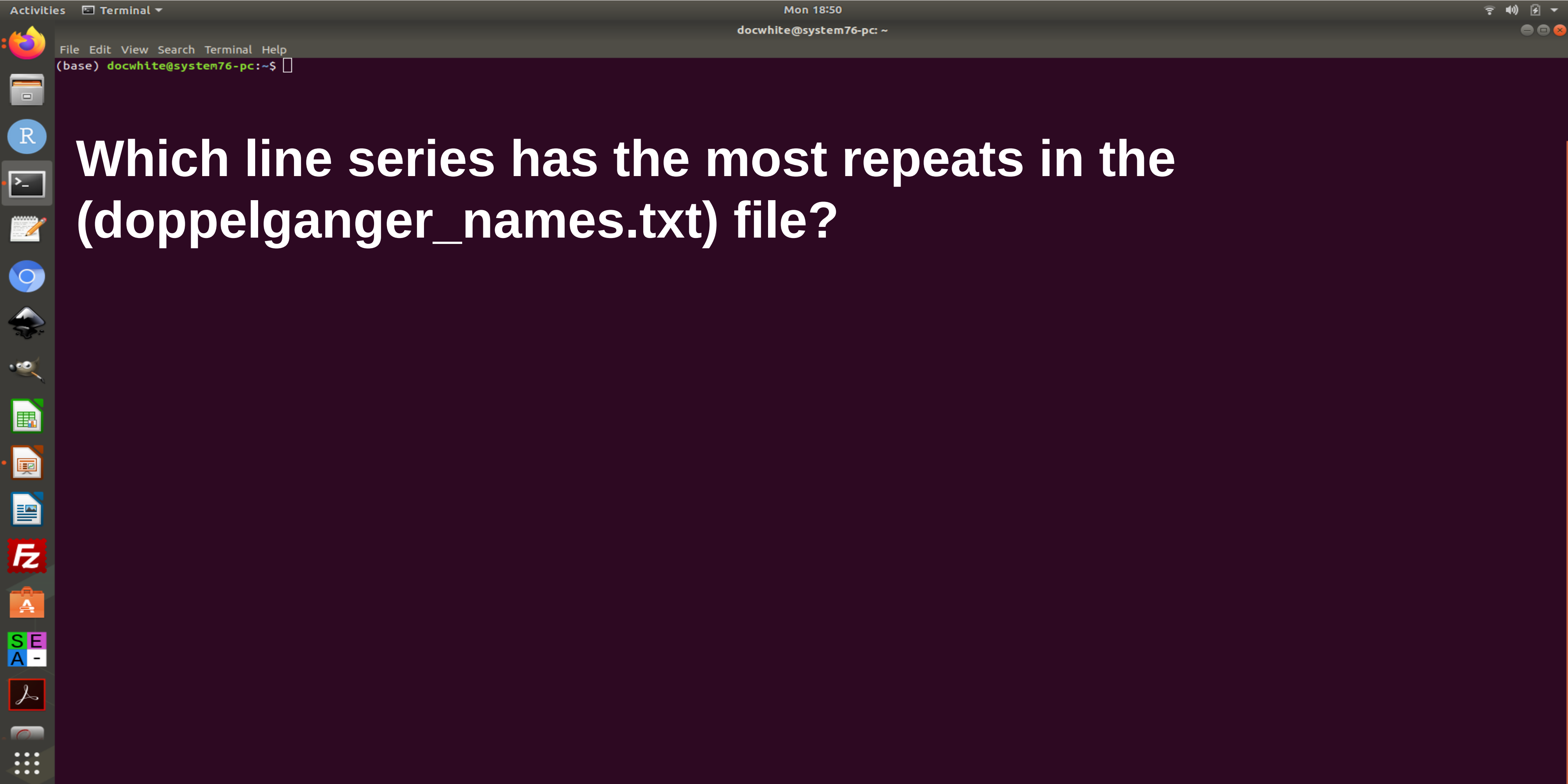
```
(base) docwhite@system76-pc:~$
```

Using the doppelganger_names.txt how many unique lines are there?

```
sort -k1 doppelganger_names.txt | uniq -c | wc -l
```

```
5
```

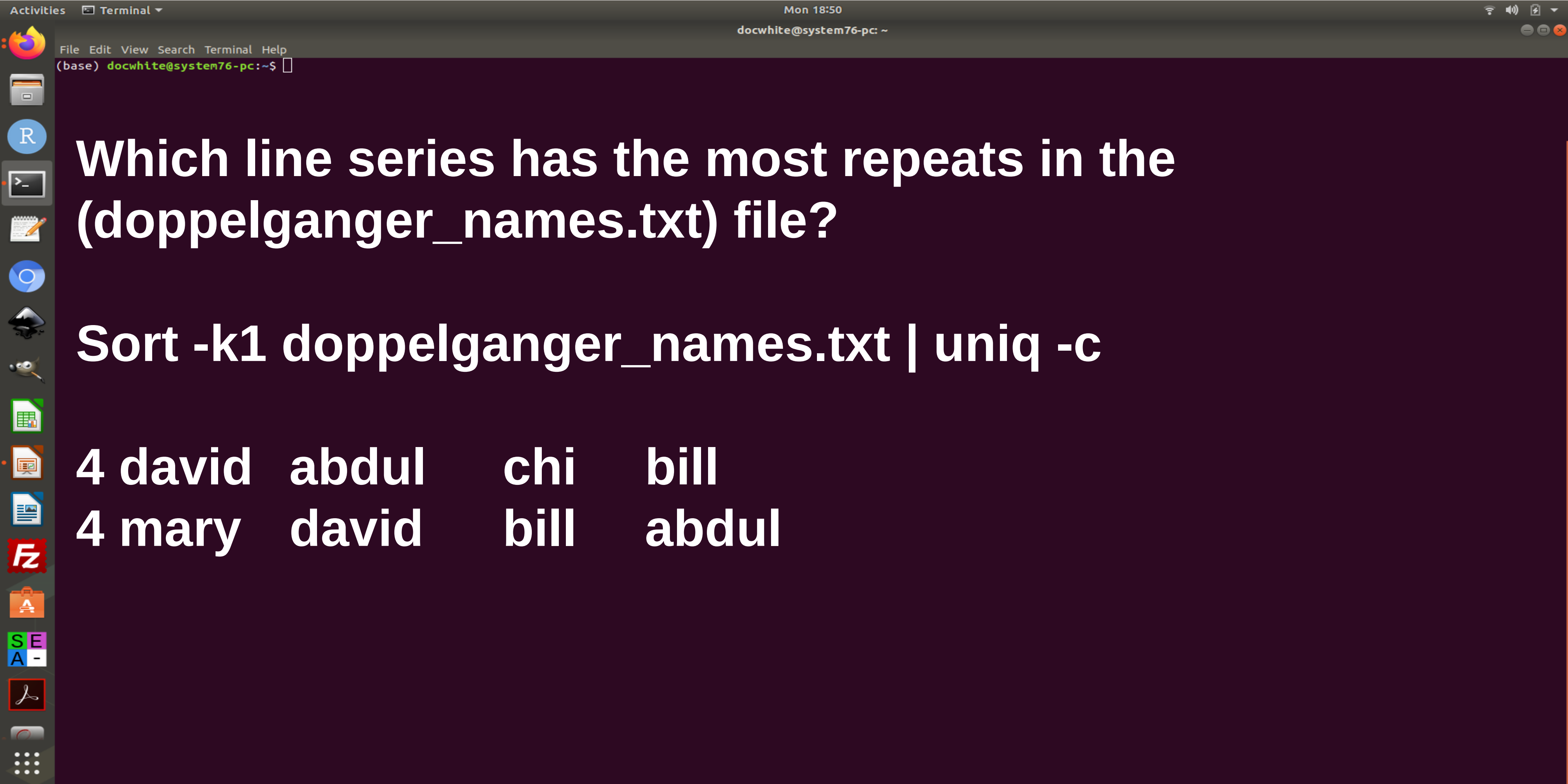
uniq examples



```
(base) docwhite@system76-pc:~$ 
```

Which line series has the most repeats in the (doppelganger_names.txt) file?

uniq examples



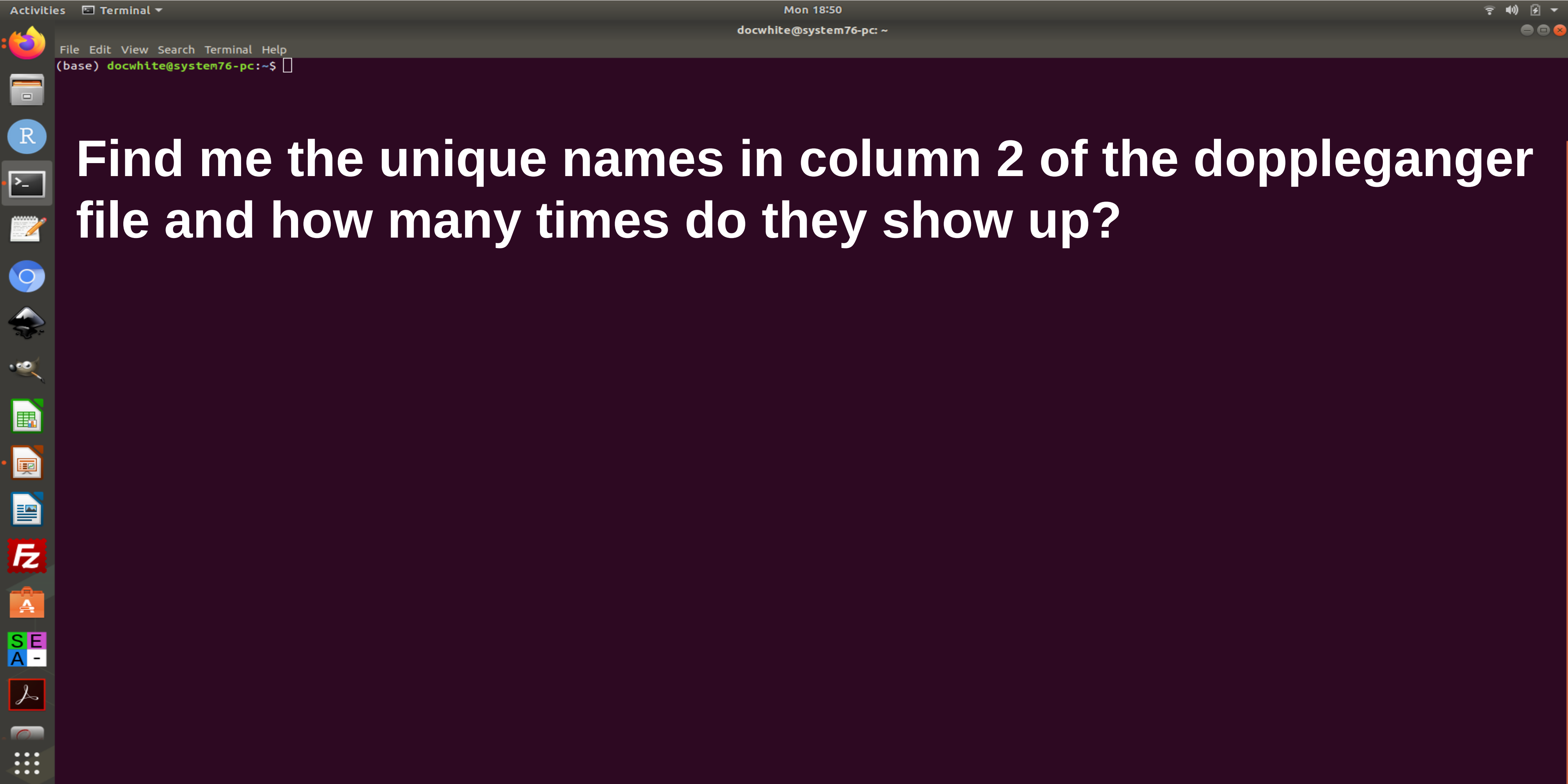
```
(base) docwhite@system76-pc:~$
```

Which line series has the most repeats in the (doppelganger_names.txt) file?

Sort -k1 doppelganger_names.txt | uniq -c

4 david abdul chi bill
4 mary david bill abdul

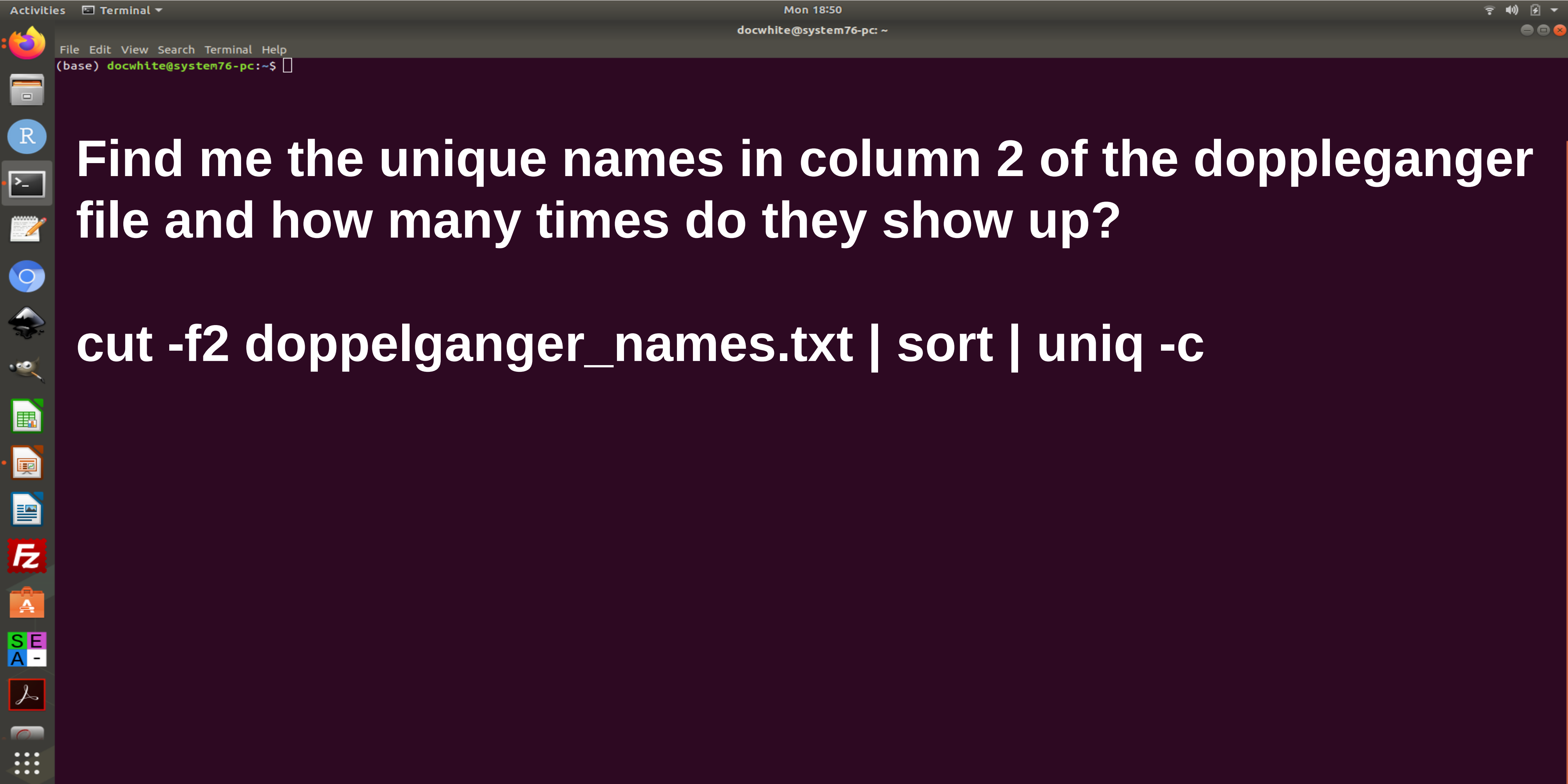
All together now examples



```
(base) docwhite@system76-pc:~$ 
```

Find me the unique names in column 2 of the doppleganger file and how many times do they show up?

All together now examples

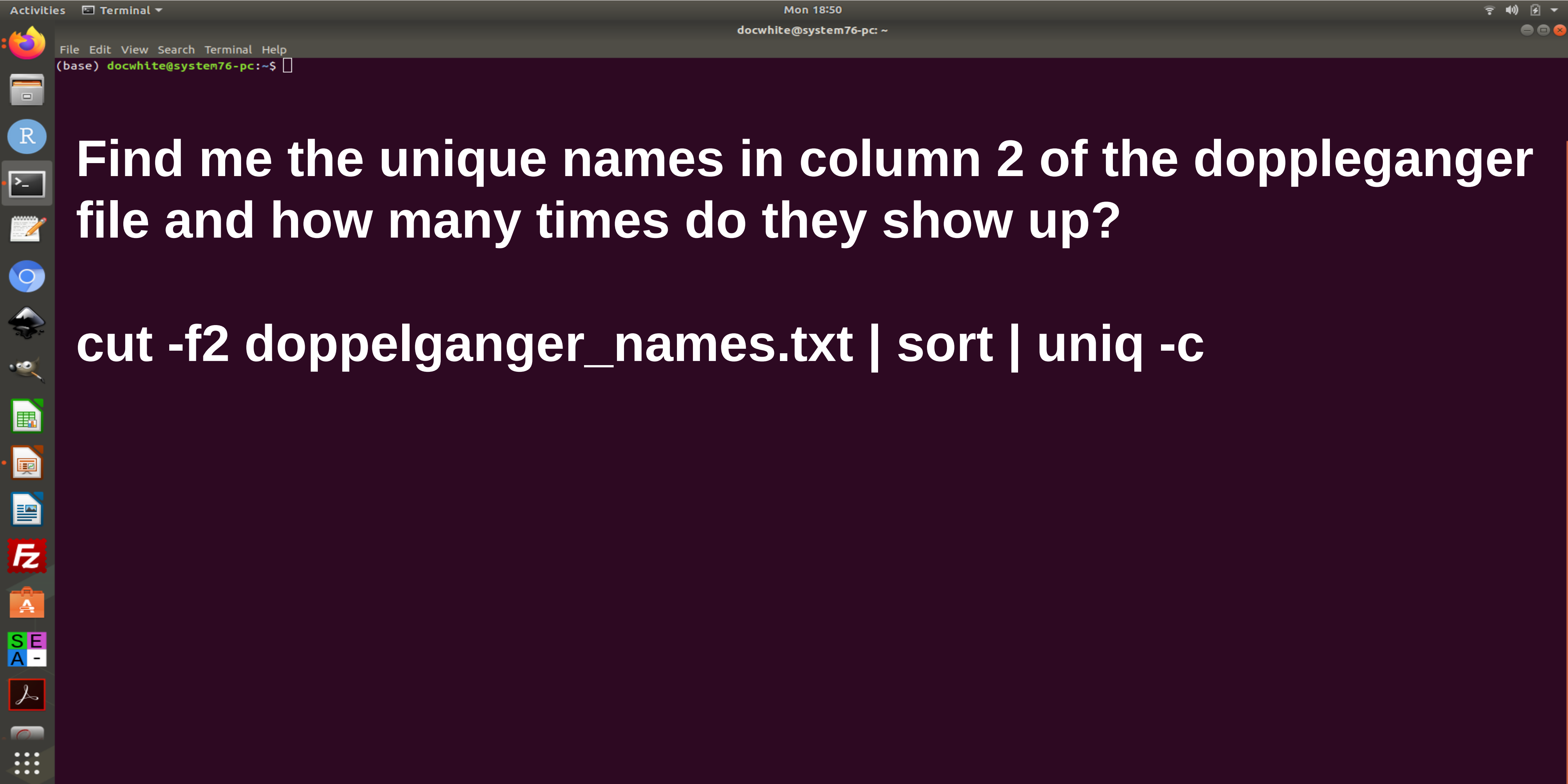


```
(base) docwhite@system76-pc:~$
```

Find me the unique names in column 2 of the doppleganger file and how many times do they show up?

```
cut -f2 doppelganger_names.txt | sort | uniq -c
```

All together now examples



```
(base) docwhite@system76-pc:~$
```

Find me the unique names in column 2 of the doppleganger file and how many times do they show up?

```
cut -f2 doppelganger_names.txt | sort | uniq -c
```

Quiz 6

- On canvas now

Bonus 5

- In the doppelganger_names.txt count how many times the name ‘chi’ is left to the name ‘bill’

Using grep only command:

Using grep with printf command:

Only awk: