BINF2111 – Introduction to Bioinformatics Computing

UNIX 101 – enter the coding zone

UNIX

Richard Allen White III, PhD
RAW Lab
Lecture 6 – Thursday Sep 7th, 2023

Learning Objectives

- Review quiz and bonus

- wget/zip/tar

- tr/printf command

- text editors

- Quiz 6

Bonus 5

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

→ tr

→ awk

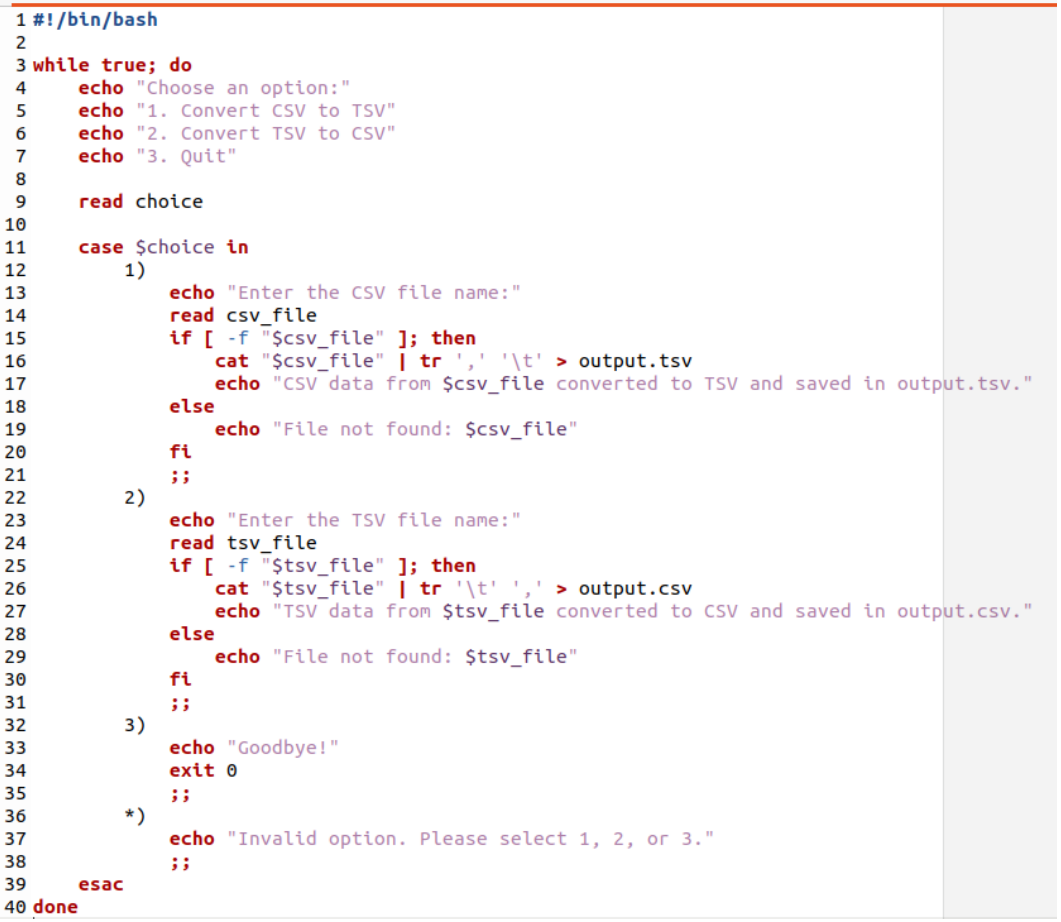
Bonus 5

- 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

Bonus 5



```bash
#!/bin/bash

while true; do
    echo "Choose an option:"
    echo "1. Convert CSV to TSV"
    echo "2. Convert TSV to CSV"
    echo "3. Quit"

    read choice

    case $choice in
        1)
            echo "Enter the CSV file name:"
            read csv_file
            if [ -f "$csv_file" ]; then
                cat "$csv_file" | tr ',' '\t' > output.tsv
                echo "CSV data from $csv_file converted to TSV and saved in output.tsv."
            else
                echo "File not found: $csv_file"
            fi
            ;;
        2)
            echo "Enter the TSV file name:"
            read tsv_file
            if [ -f "$tsv_file" ]; then
                cat "$tsv_file" | tr '\t' ',' > output.csv
                echo "TSV data from $tsv_file converted to CSV and saved in output.csv."
            else
                echo "File not found: $tsv_file"
            fi
            ;;
        3)
            echo "Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid option. Please select 1, 2, or 3."
            ;;
    esac
done
```

BASH

```rust
use std::fs::File;
use std::io::{self, BufRead, BufReader, Write};

fn main() {
    loop {
        println!("Choose an option:");
        println!("1. Convert CSV to TSV");
        println!("2. Convert TSV to CSV");
        println!("3. Quit");

        let mut choice = String::new();
        io::stdin()
            .read_line(&mut choice)
            .expect("Failed to read choice");

        match choice.trim() {
            "1" => {
                println!("Enter the CSV file name:");
                let mut csv_file = String::new();
                io::stdin()
                    .read_line(&mut csv_file)
                    .expect("Failed to read file name");
                let csv_file = csv_file.trim();

                match File::open(&csv_file) {
                    Ok(file) => {
                        let tsv_file = "output.tsv";
                        let output_file = File::create(tsv_file).expect("Failed to create output file");

                        let mut writer = io::BufWriter::new(output_file);
                        let reader = BufReader::new(file);

                        for line in reader.lines() {
                            let line = line.expect("Failed to read line");
                            let tsv_line = line.replace(",", "\t");
                            writer.write_all(tsv_line.as_bytes()).expect("Failed to write to output file");
                            writer.write_all(b"\n").expect("Failed to write to output file");
                        }

                        println!("CSV data from {} converted to TSV and saved in {}.", csv_file, tsv_file);
                    }
                    Err(_) => {
                        println!("File not found: {}", csv_file);
                    }
                }
            }
            "2" => {
                println!("Enter the TSV file name:");
                let mut tsv_file = String::new();
                io::stdin()
                    .read_line(&mut tsv_file)
                    .expect("Failed to read file name");
                let tsv_file = tsv_file.trim();

                match File::open(&tsv_file) {
                    Ok(file) => {
                        let csv_file = "output.csv";
                        let output_file = File::create(csv_file).expect("Failed to create output file");

                        let mut writer = io::BufWriter::new(output_file);
                        let reader = BufReader::new(file);

                        for line in reader.lines() {
                            let line = line.expect("Failed to read line");
                            let csv_line = line.replace("\t", ",");
                            writer.write_all(csv_line.as_bytes()).expect("Failed to write to output file");
                            writer.write_all(b"\n").expect("Failed to write to output file");
                        }

                        println!("TSV data from {} converted to CSV and saved in {}.", tsv_file, csv_file);
                    }
                    Err(_) => {
                        println!("File not found: {}", tsv_file);
                    }
                }
            }
            "3" => {
                println!("Goodbye!");
                break;
            }
            _ => {
                println!("Invalid option. Please select 1, 2, or 3.");
            }
        }
    }
}
```

Rust

Missed Quiz Question

```
echo $PATH
```

Means?

Missed Quiz Question

echo $PATH

Means?

lists all directories

Missed Quiz Question

echo $PATH

Means?

lists all directories - NO (ls)

Missed Quiz Question

echo $PATH

Means?

lists all directories - NO (ls)

writes working directory?

Missed Quiz Question

echo $PATH

Means?

lists all directories - NO (ls)

writes working directory - NO

Missed Quiz Question

echo $PATH

Means?

lists all directories - NO (ls)

writes working directory - NO (pwd)

Missed Quiz Question

echo $PATH

Means?

lists all directories - NO (ls)

writes working directory - NO (pwd)

writes your path list of EXECUTABLE files

Missed Quiz Question

What will these commands do?

LINUX

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

MAC

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

A] cuts columns 1 and 4, doesn't print

B] prints all but columns 1 and 4, within tabs

C] print all but columns 1 and 4

Missed Quiz Question

What will these commands do?

LINUX

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

MAC

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

Why doesn’t answer A work?

It’s true right?

A] cuts columns 1 and 4, doesn't print

B] prints all but columns 1 and 4, within tabs

C] print all but columns 1 and 4

Missed Quiz Question

What will these commands do?

LINUX

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

MAC

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

A] cuts columns 1 and 4, doesn't print

B] prints all but columns 1 and 4, within tabs

C] print all but columns 1 and 4

Why doesn’t answer A work?

It’s true right? - WRONG

Missed Quiz Question

What will these commands do?

LINUX

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

MAC

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

It does print columns 2 and 3
BUT NOT column 1 and 4.

A] cuts columns 1 and 4, doesn't print

B] prints all but columns 1 and 4, within tabs

C] print all but columns 1 and 4

Missed Quiz Question

What will this command do?

```
egrep -v '^$' file.tsv
```

A] cuts columns 1 and 4, doesn't print

B] delete all empty lines

C] print all but columns 1 and 4

D] cut all empty lines

Missed Quiz Question

What will this command do?

```
egrep -v '^$' file.tsv
```

A] cuts columns 1 and 4, doesn't print

B] delete all empty lines

C] print all but columns 1 and 4

D] cut all empty lines

Missed Quiz Question

If I had a file with lots of repeated lines who would I count the unique lines?

A] sort -k1 file.txt | uniq -u

B] sort -k1 file.txt | uniq -w

C] sort -k1 file.txt | uniq -c1

D] sort -k1 file.txt | uniq -c

Missed Quiz Question

If I had a file with lots of repeated lines who would I count the unique lines?

A] sort -k1 file.txt | uniq -u

B] sort -k1 file.txt | uniq -w

C] sort -k1 file.txt | uniq -c1

D] sort -k1 file.txt | uniq -c

-c counts, -u only prints unique lines

Missed Quiz Question

One of these commands will not remove empty lines/white space

A] grep -v -e '^[[:space:]]*$' file

B] awk 'NF > 0' file

C] sed '/^[[:space:]]*$/d' file

D] sed 's/ //g' file

Missed Quiz Question

One of these commands will not remove empty lines/white space

A] grep -v -e '^[[:space:]]*$' file

B] awk 'NF > 0' file

C] sed '/^[[:space:]]*$/d' file

D] sed 's/ //g' file - Why?

Missed Quiz Question

One of these commands will not remove empty lines/white space

A] grep -v -e '^[[:space:]]*$' file

B] awk 'NF > 0' file

C] sed '/^[[:space:]]*$/d' file

D] sed 's/ //g' file - only removes spaces! Not empty lines.

wget

wget [options] file.txt

-V/--version

Display the version of Wget.

-h/--help

Print a help message describing all of Wget's command-line options.

-b/--background

Go to background immediately after startup. If no output file is specified via the -o, output is redirected to wget-log.

-e/--execute command

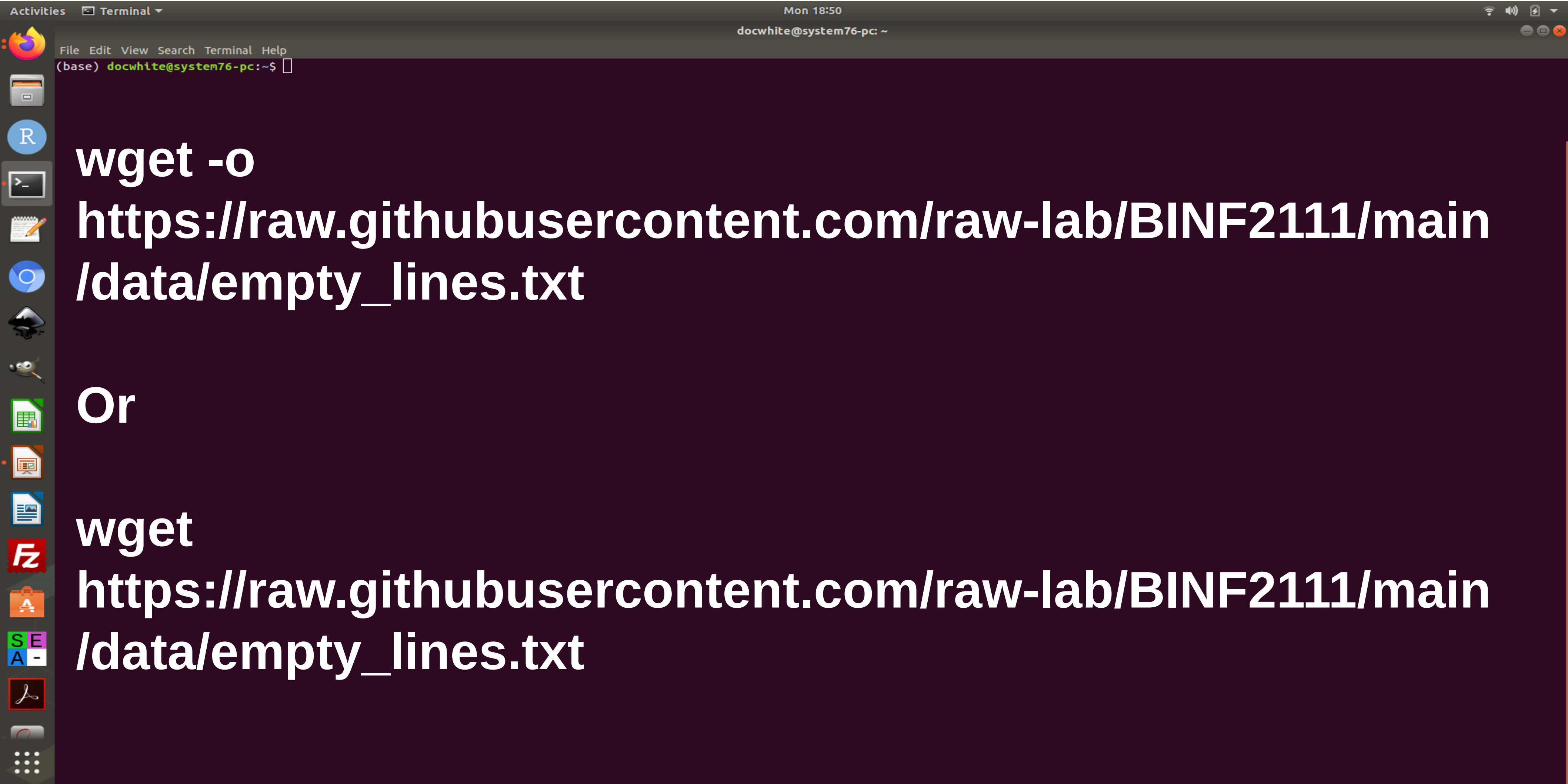
Execute command as if it were a part of .wgetrc. A command thus invoked will be executed after the commands in .wgetrc, thus taking precedence over them.

wget

wget [options] file.txt

Downloads the file to your current working directory not root or else where unless you ask it to.

wget examples



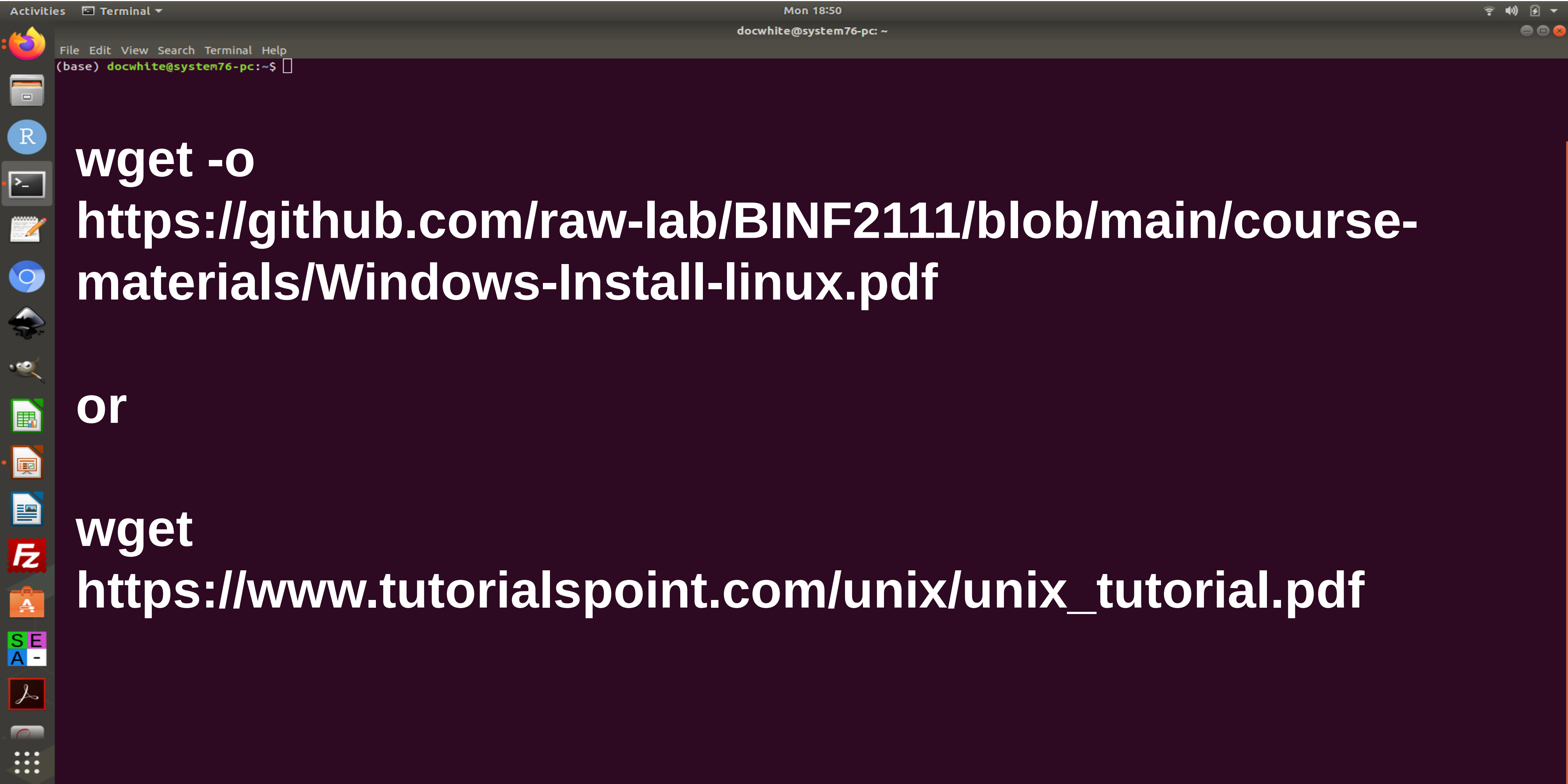
```
(base) docwhite@system76-pc:~$ 
```

```
wget -o https://raw.githubusercontent.com/raw-lab/BINF2111/main/data/empty_lines.txt
```

Or

```
wget https://raw.githubusercontent.com/raw-lab/BINF2111/main/data/empty_lines.txt
```

wget examples



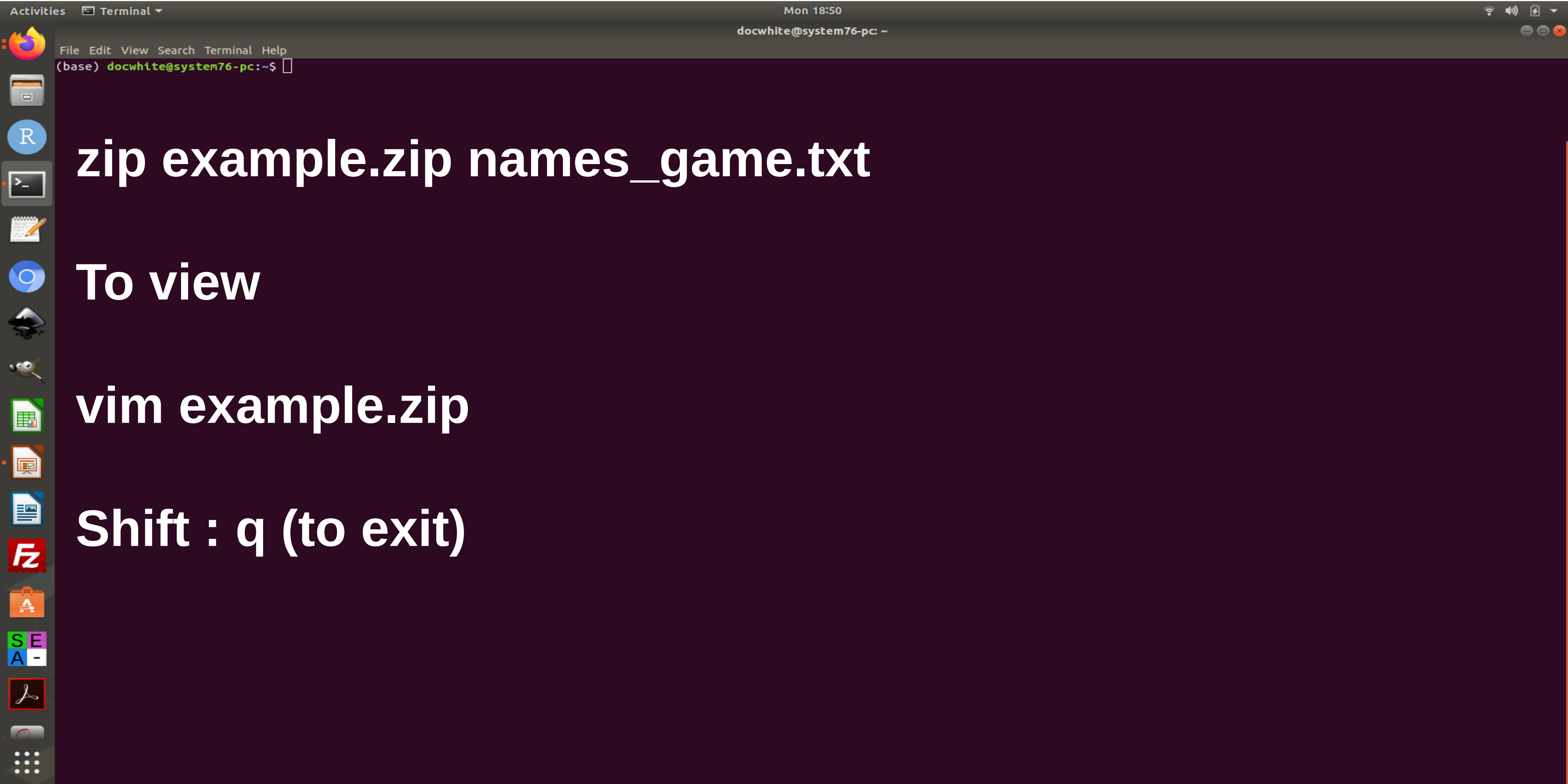
```
(base) docwhite@system76-pc:~$
```

wget -o https://github.com/raw-lab/BINF2111/blob/main/course-materials/Windows-Install-linux.pdf

or

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

zip examples



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

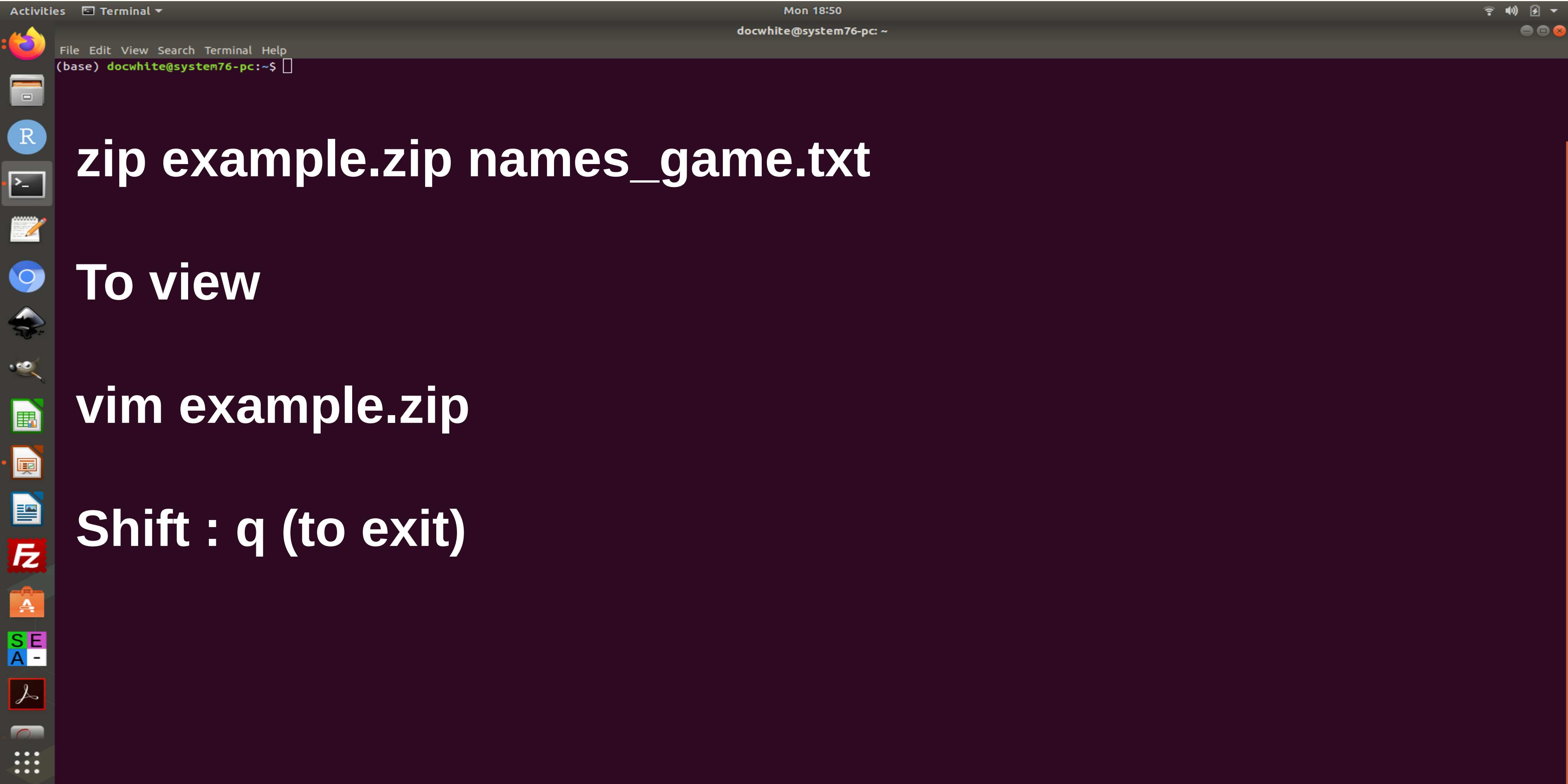
zip example.zip names_game.txt

To view

vim example.zip

Shift : q (to exit)

zip examples



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

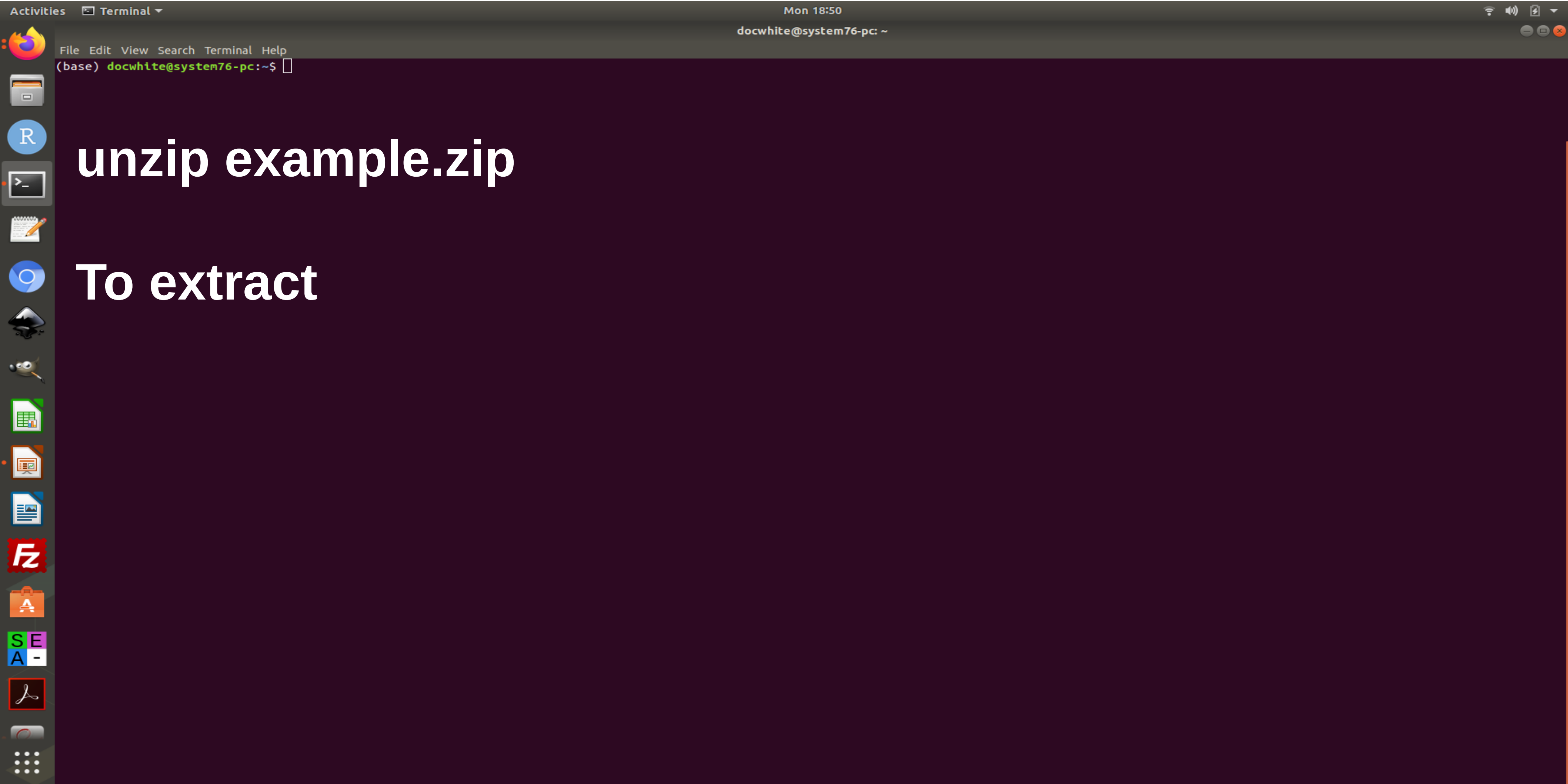
zip example.zip names_game.txt

To view

vim example.zip

Shift : q (to exit)

zip examples

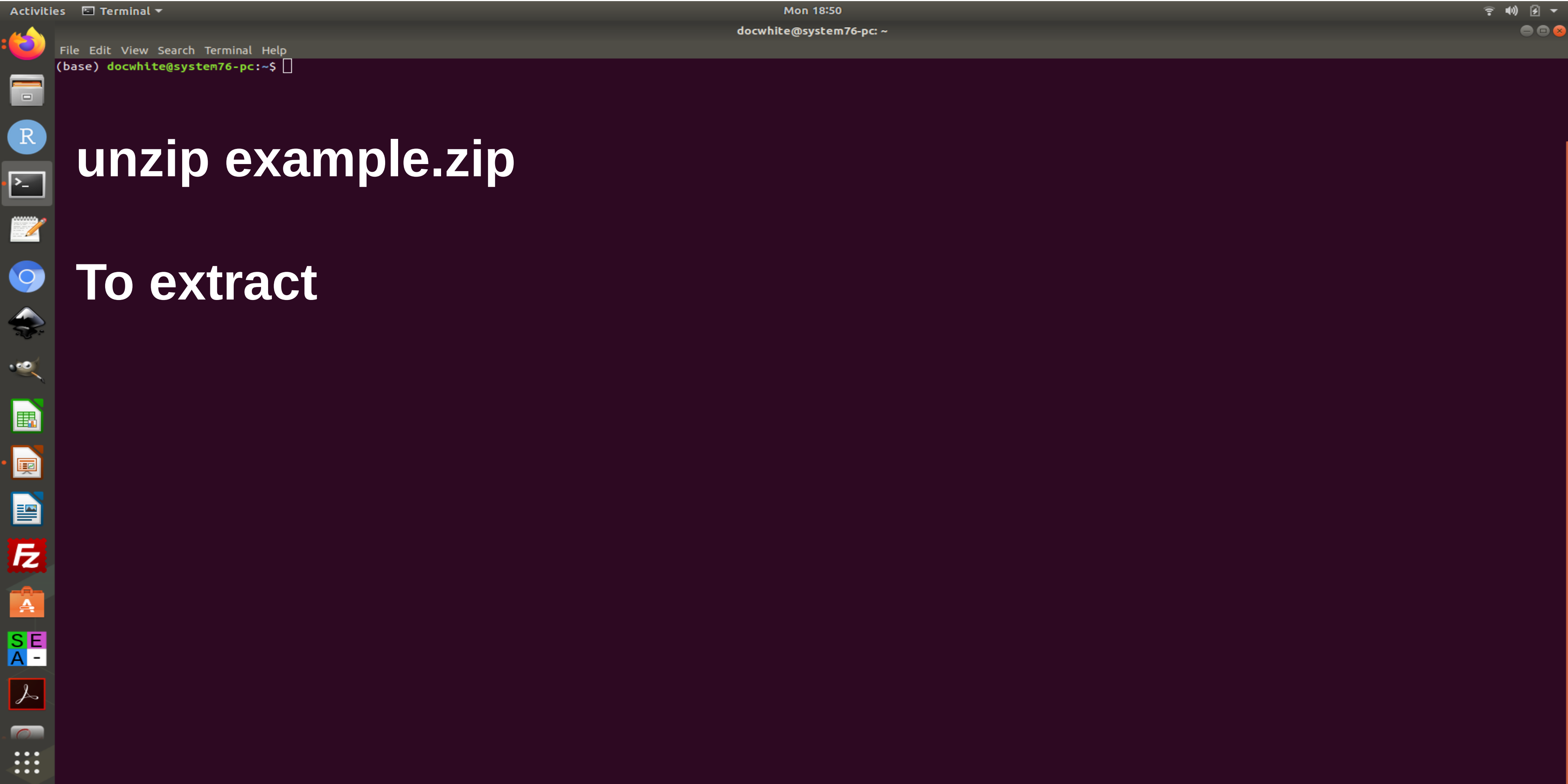


```
(base) docwhite@system76-pc:~$
```

```
unzip example.zip
```

To extract

zip examples

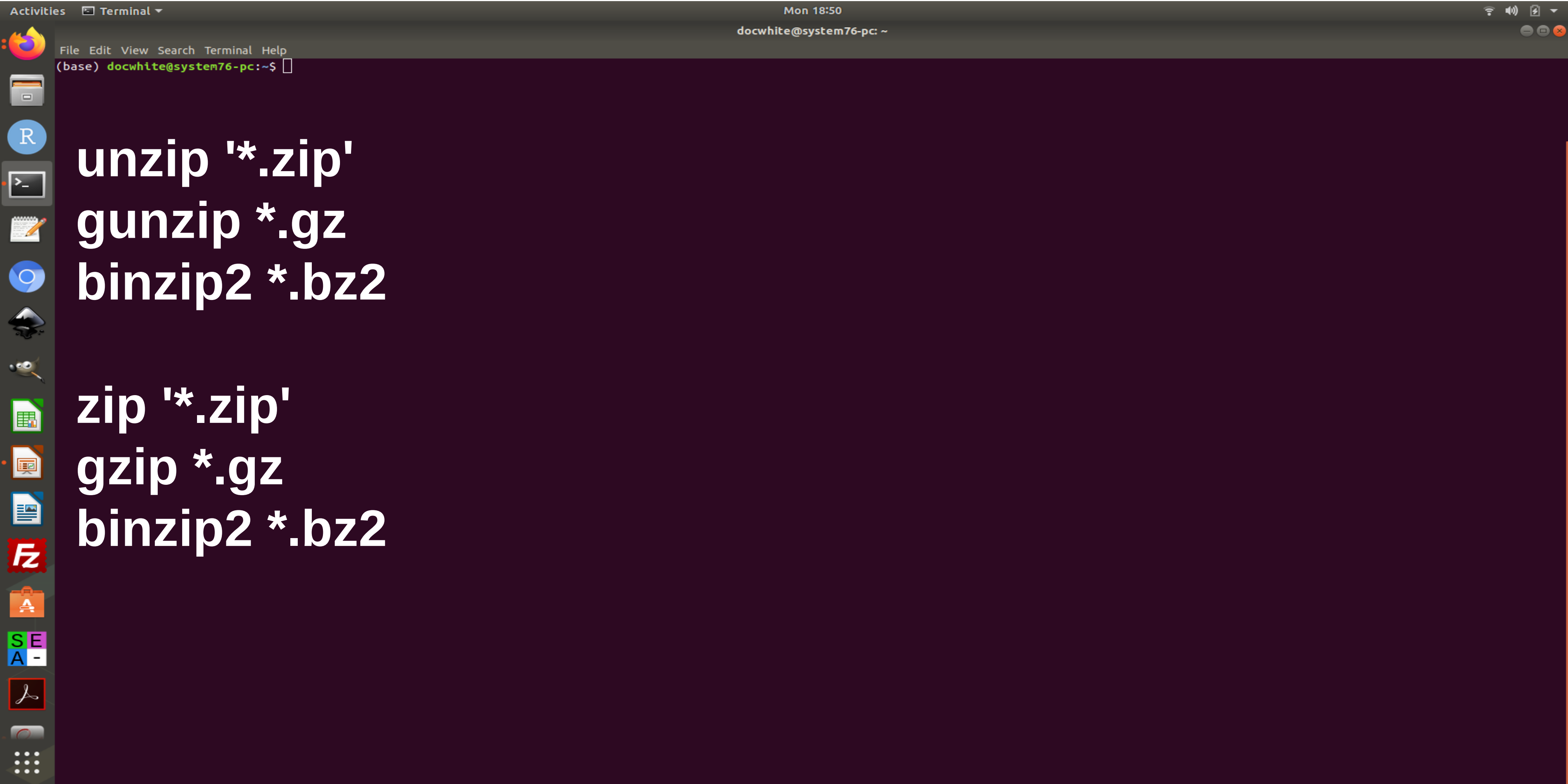


```
(base) docwhite@system76-pc:~$ 
```

unzip example.zip

To extract

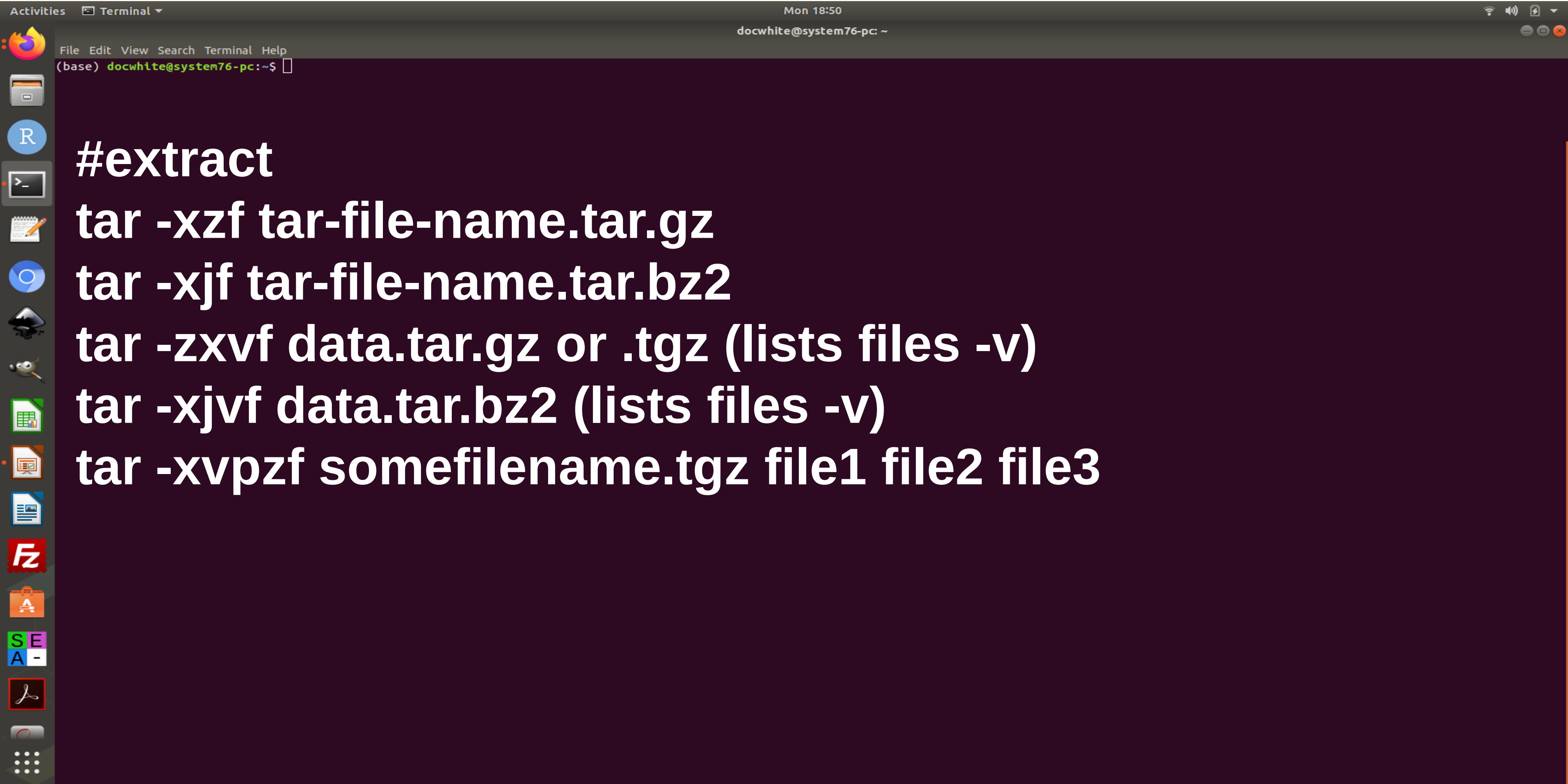
zip examples



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

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

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

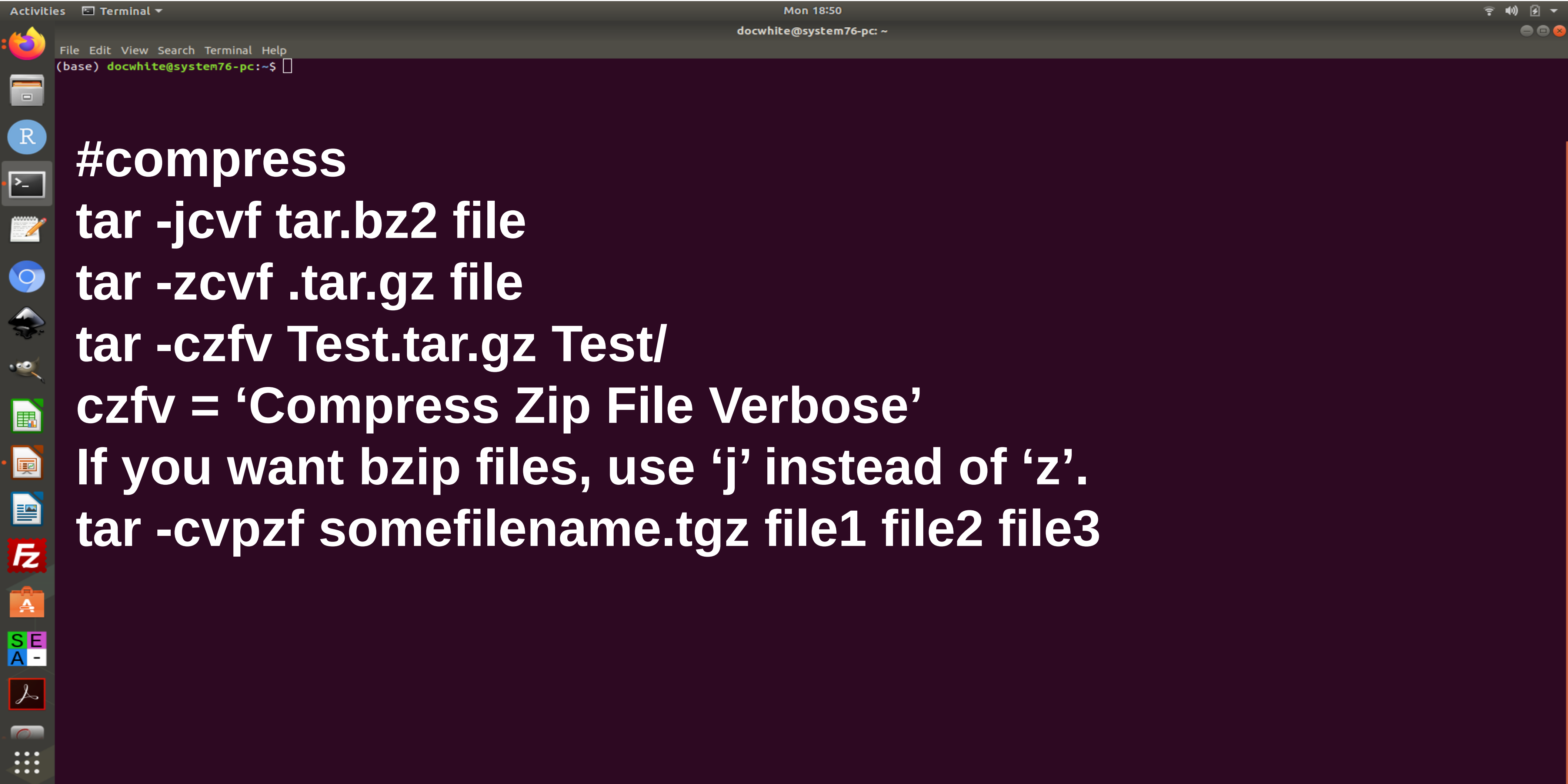
tar examples



```
(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

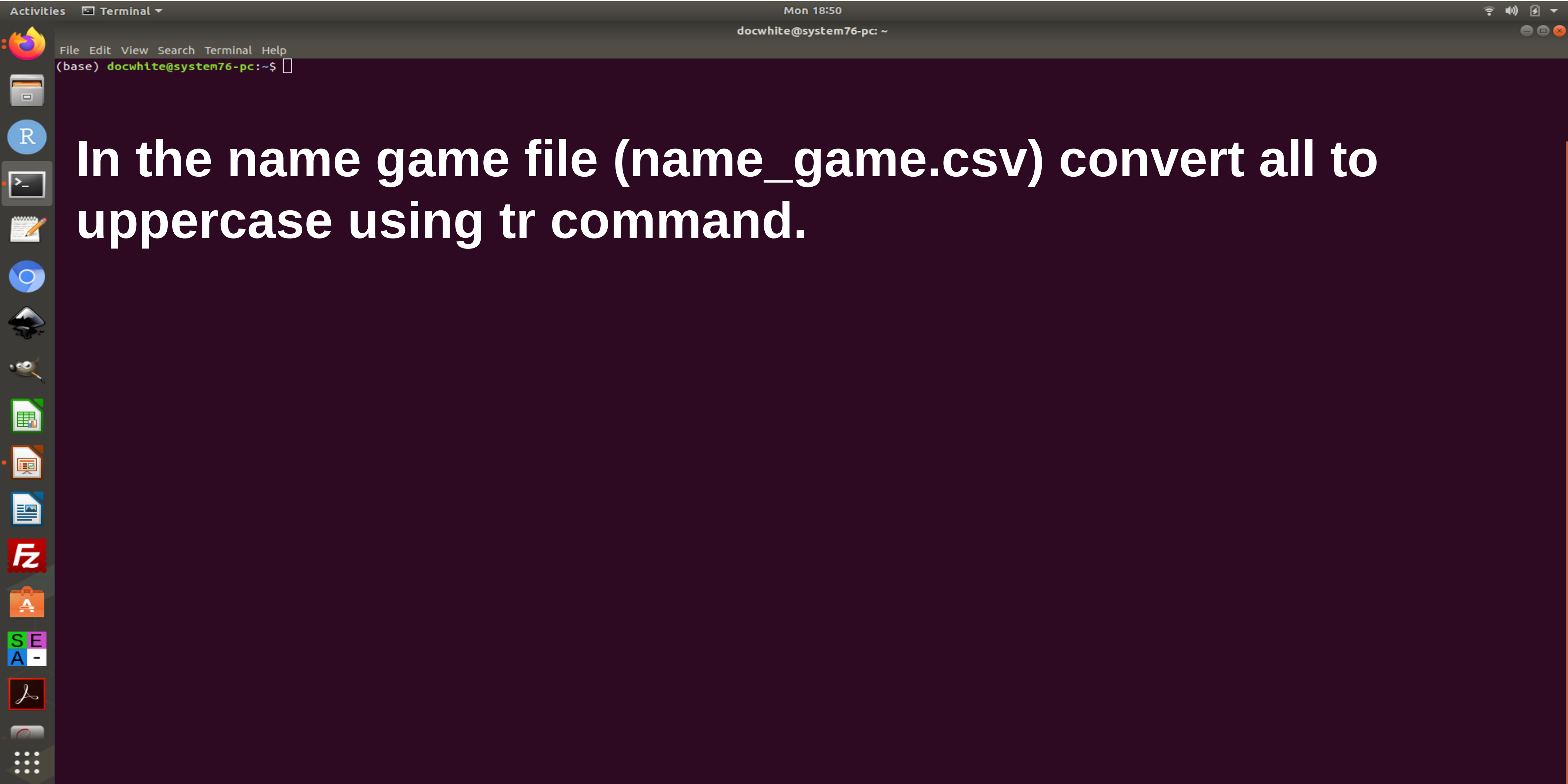
Tr (translate) – syntax anatomy UNIX tool

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]

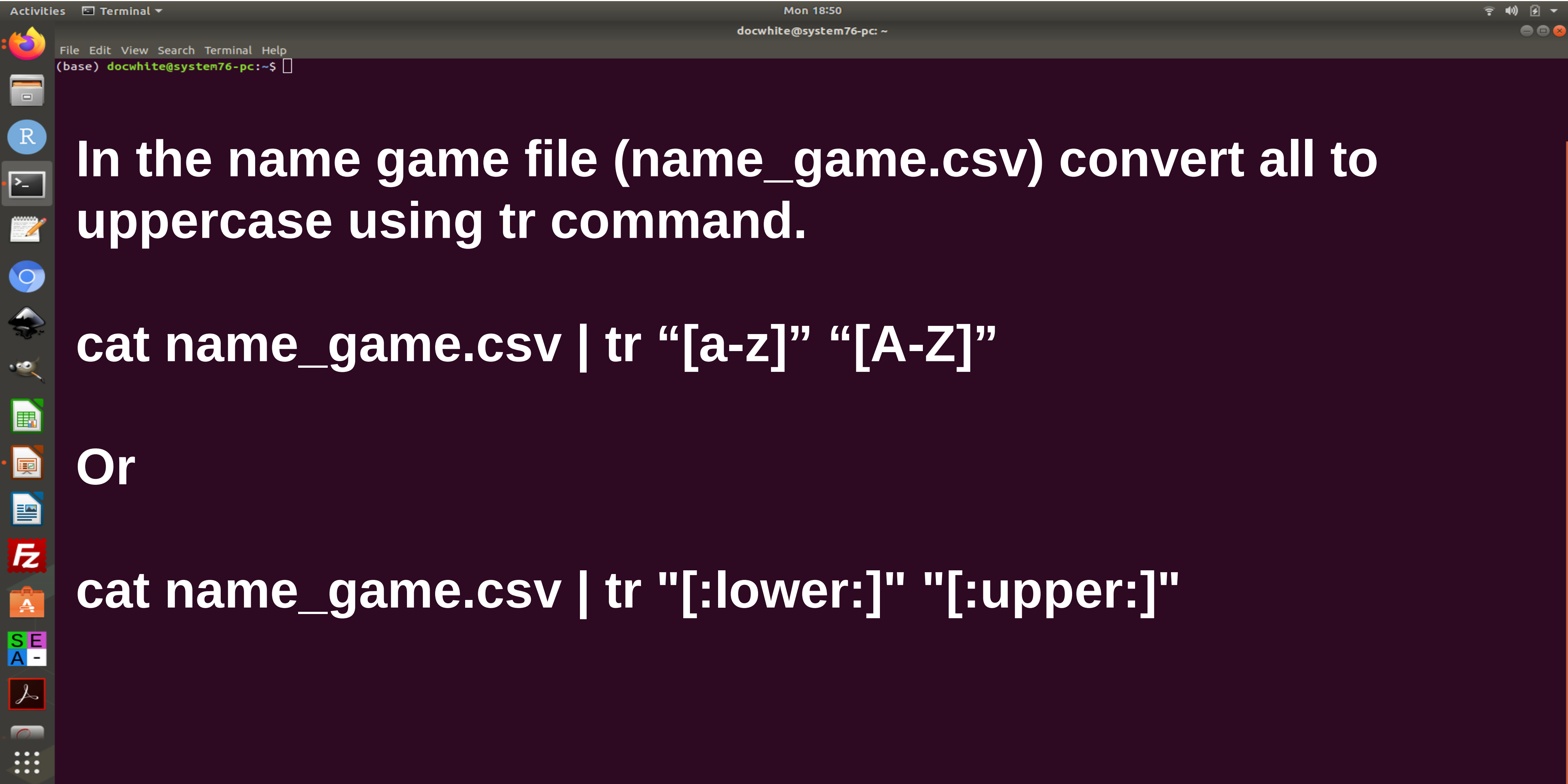
tr examples



```
(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:~$
```

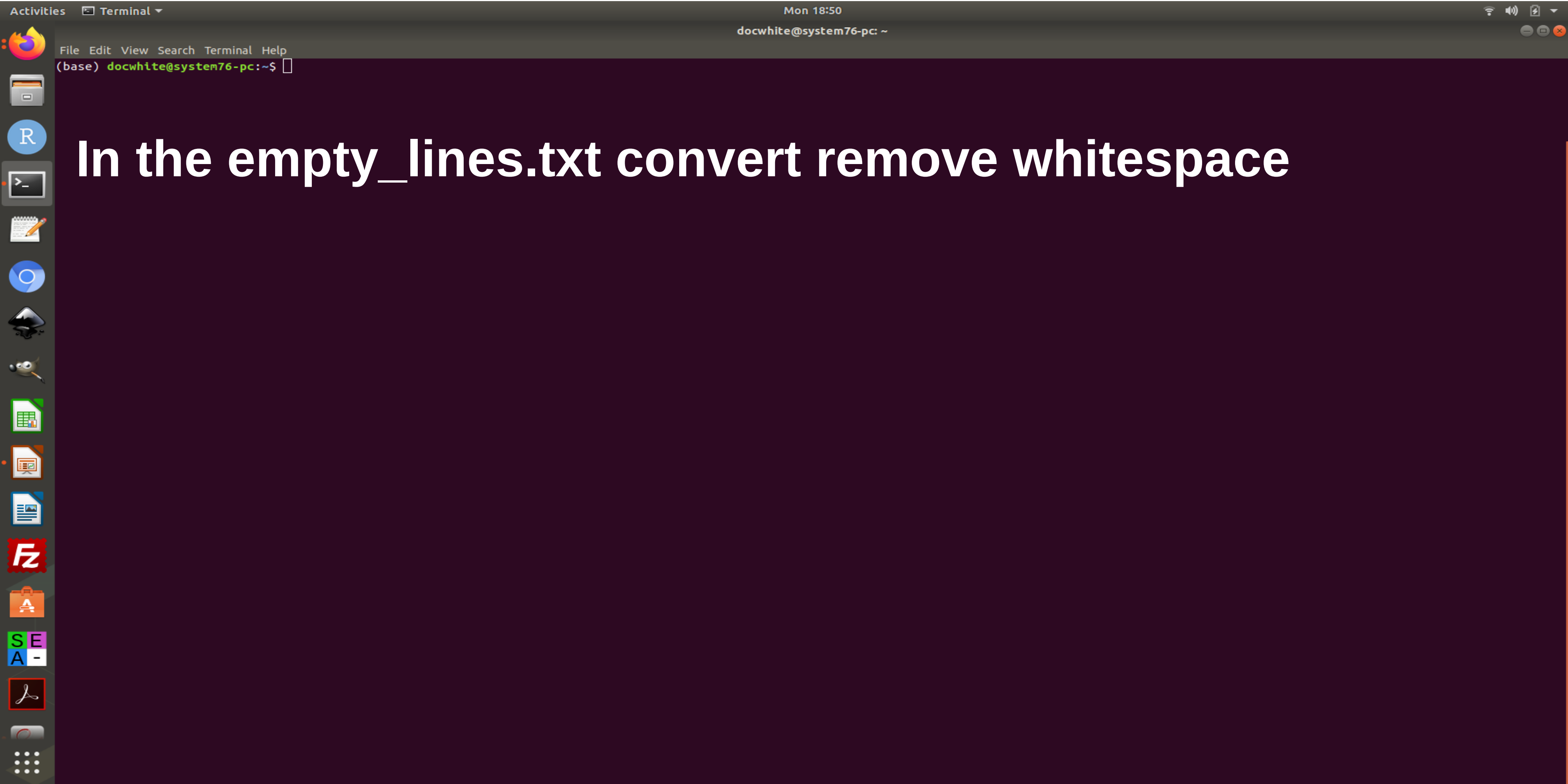
In the name game file (name_game.csv) convert all to uppercase using tr command.

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

Or

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

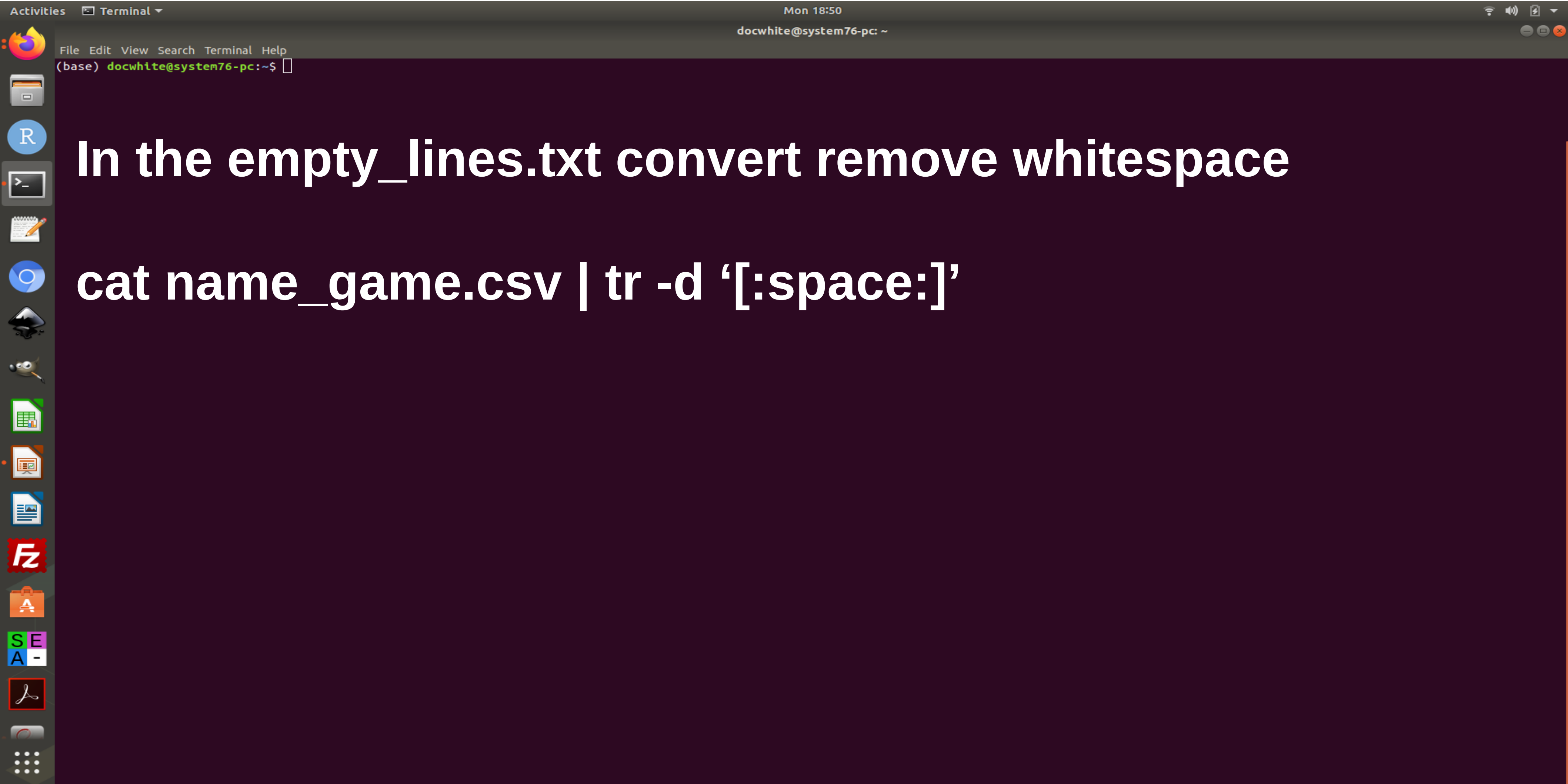
tr examples



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

In the empty_lines.txt convert remove whitespace

tr examples

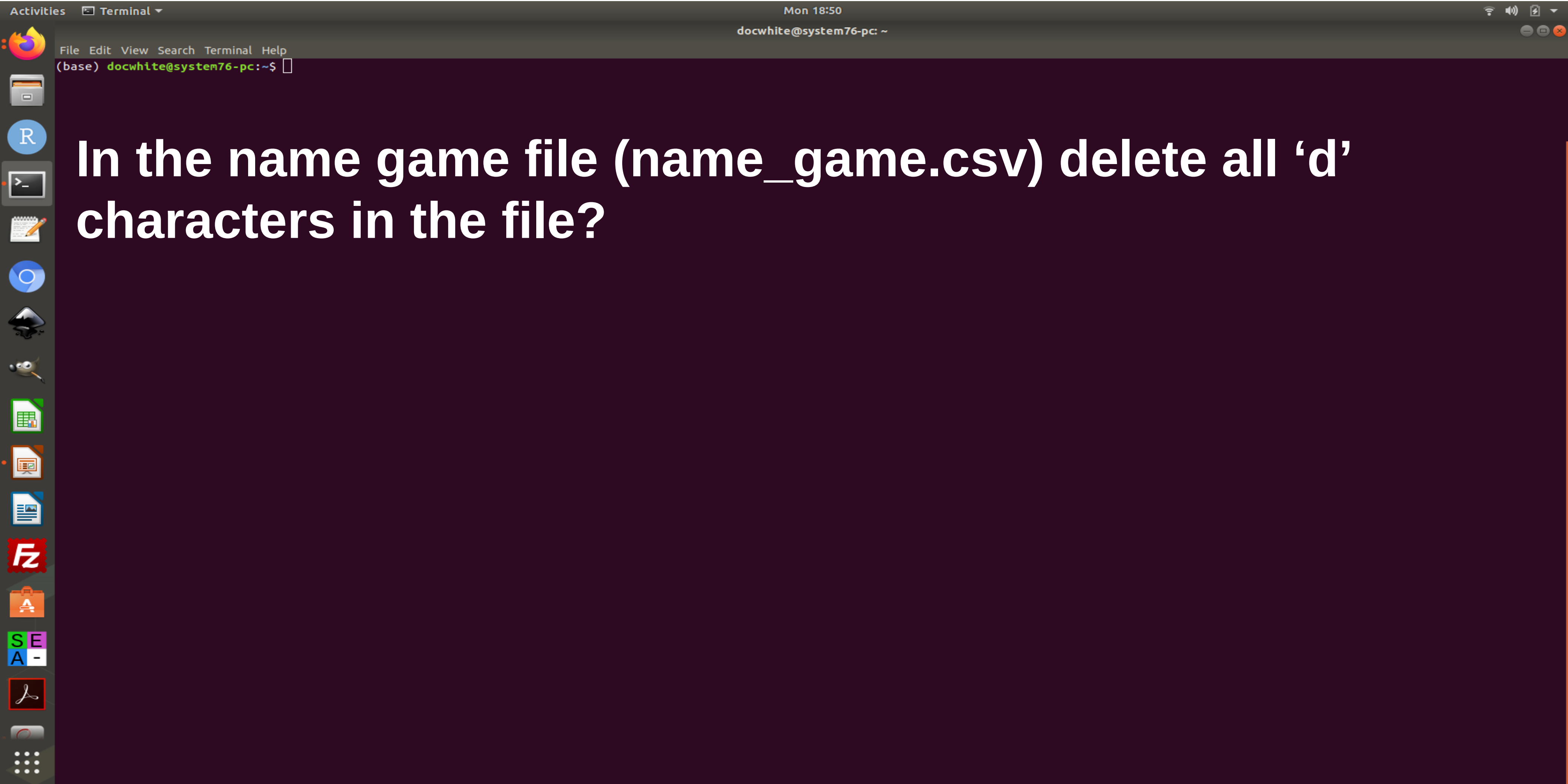


```
(base) docwhite@system76-pc:~$ 
```

In the empty_lines.txt convert remove whitespace

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

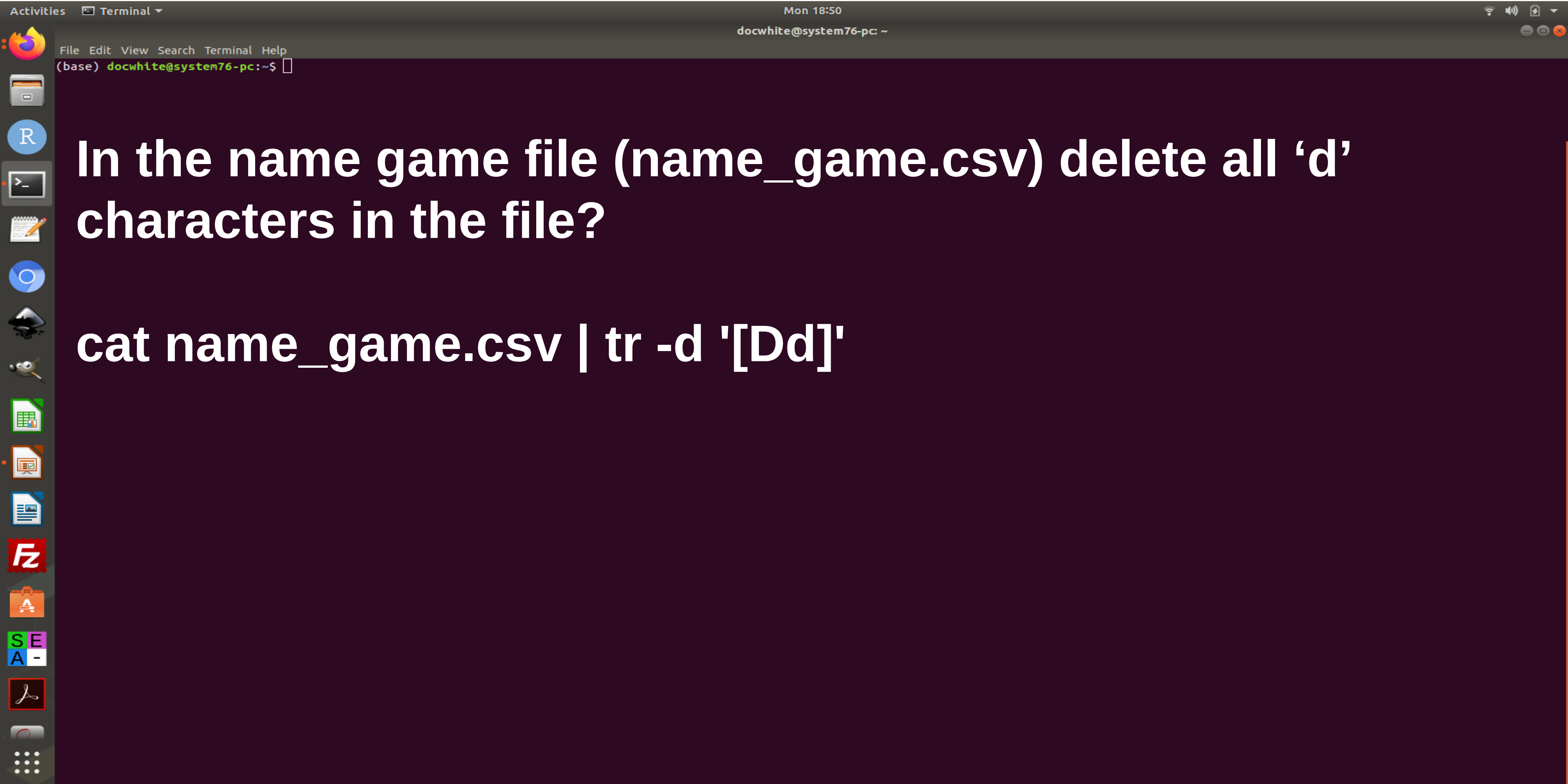
tr examples



```
(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.

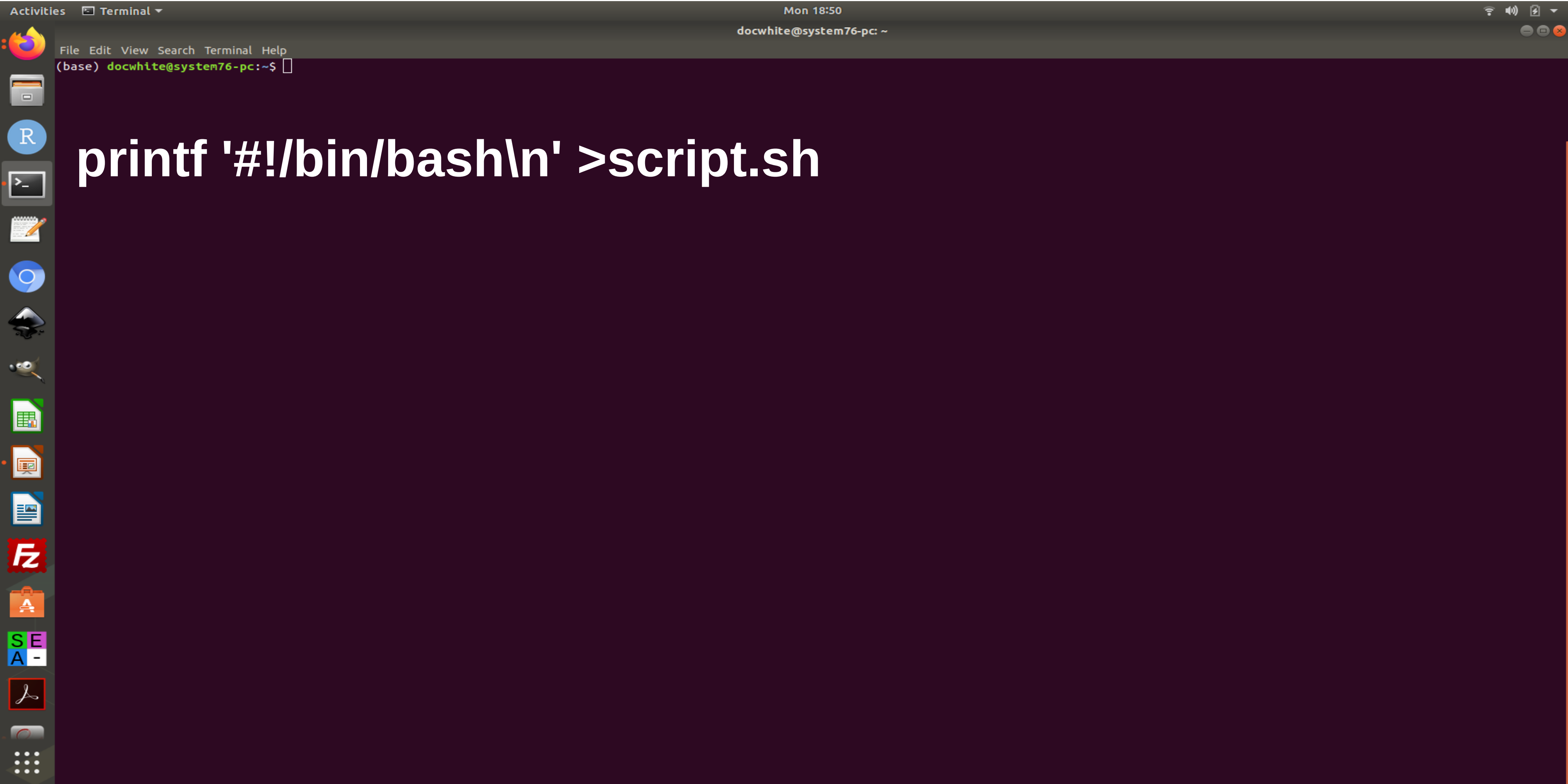
- To left align output in the field. (Default: Right align)

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.

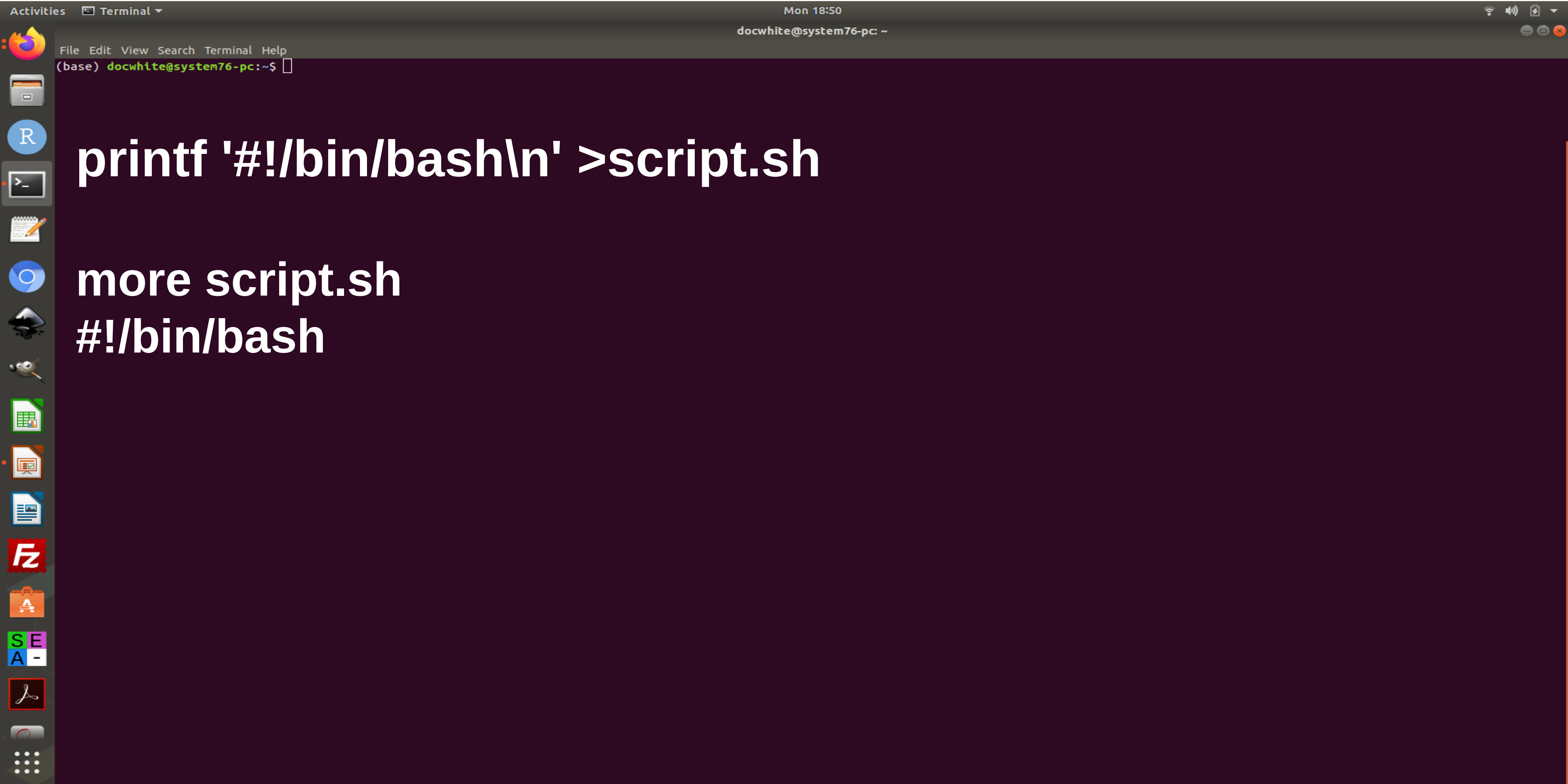
printf examples



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

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

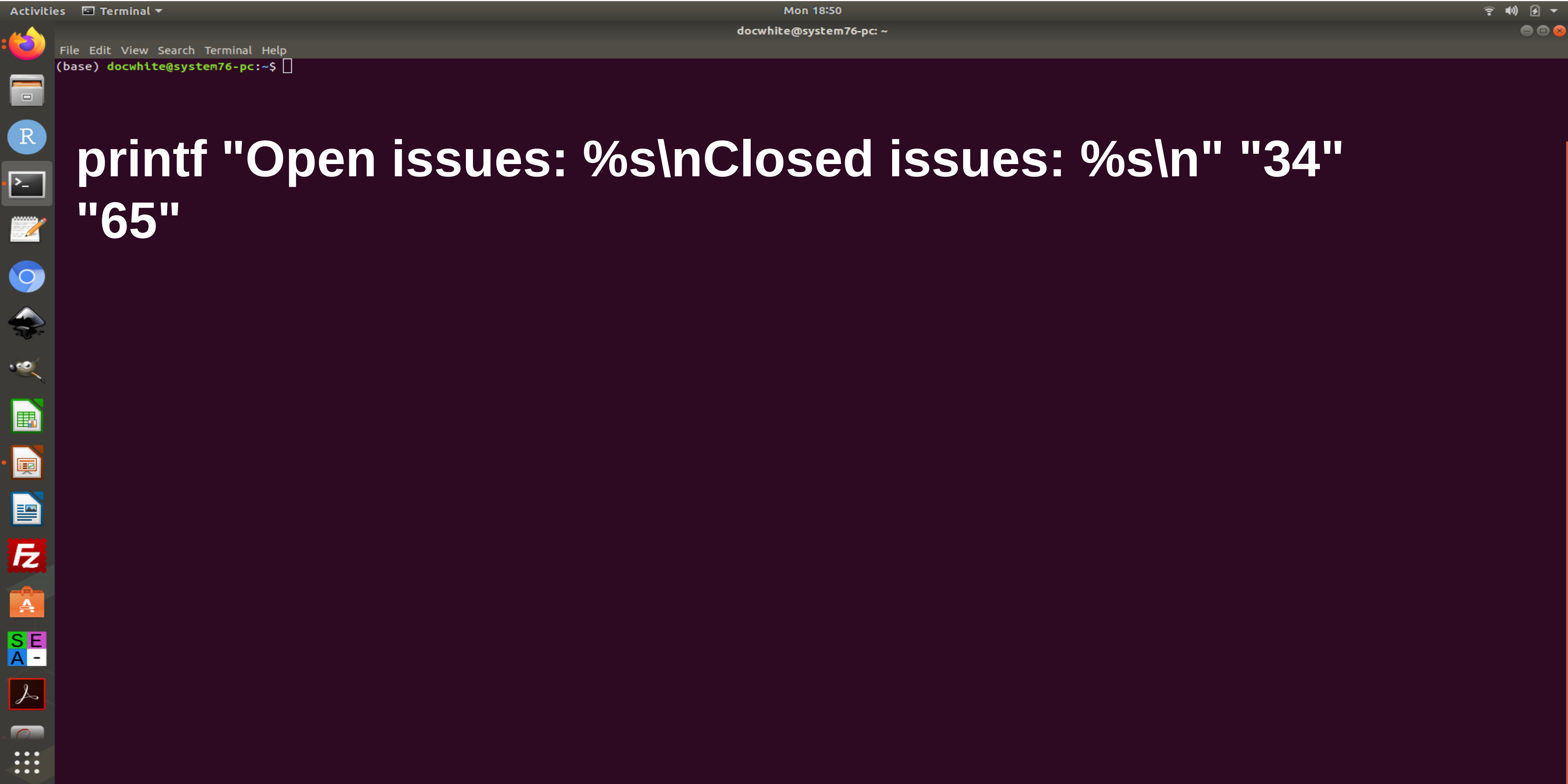
printf examples



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

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

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

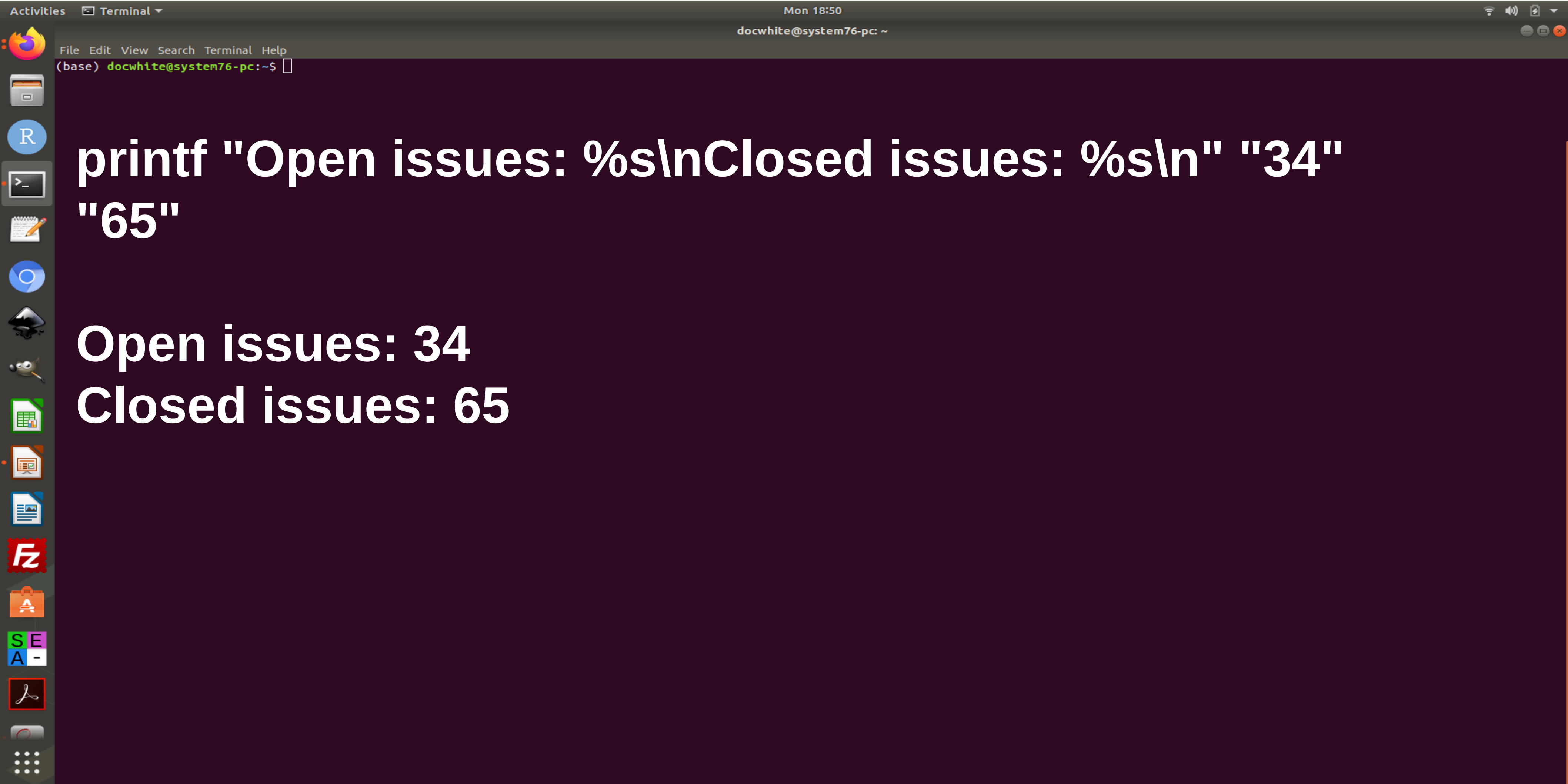
printf examples



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

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

printf examples

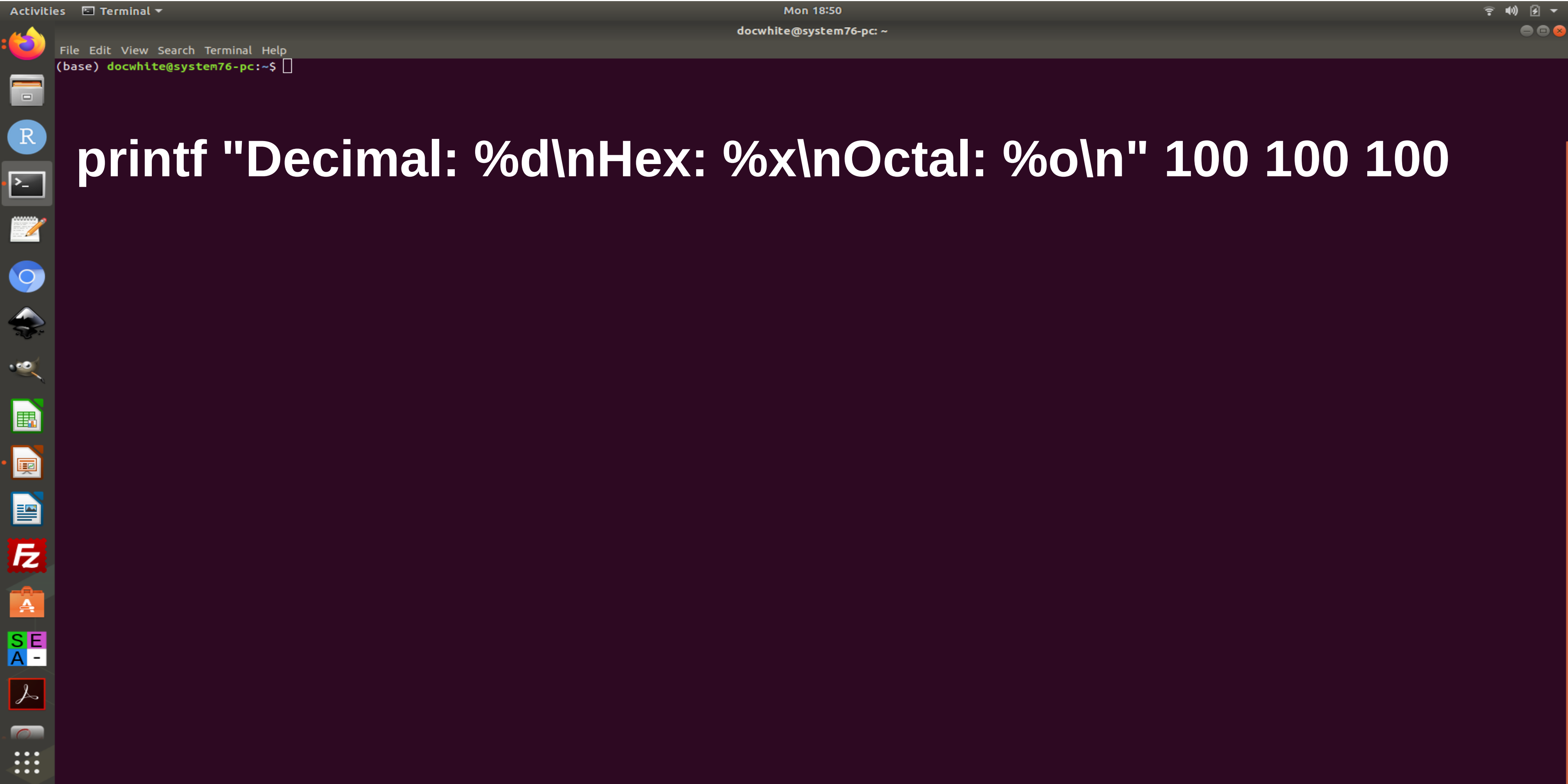


```
(base) docwhite@system76-pc:~$
```

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

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

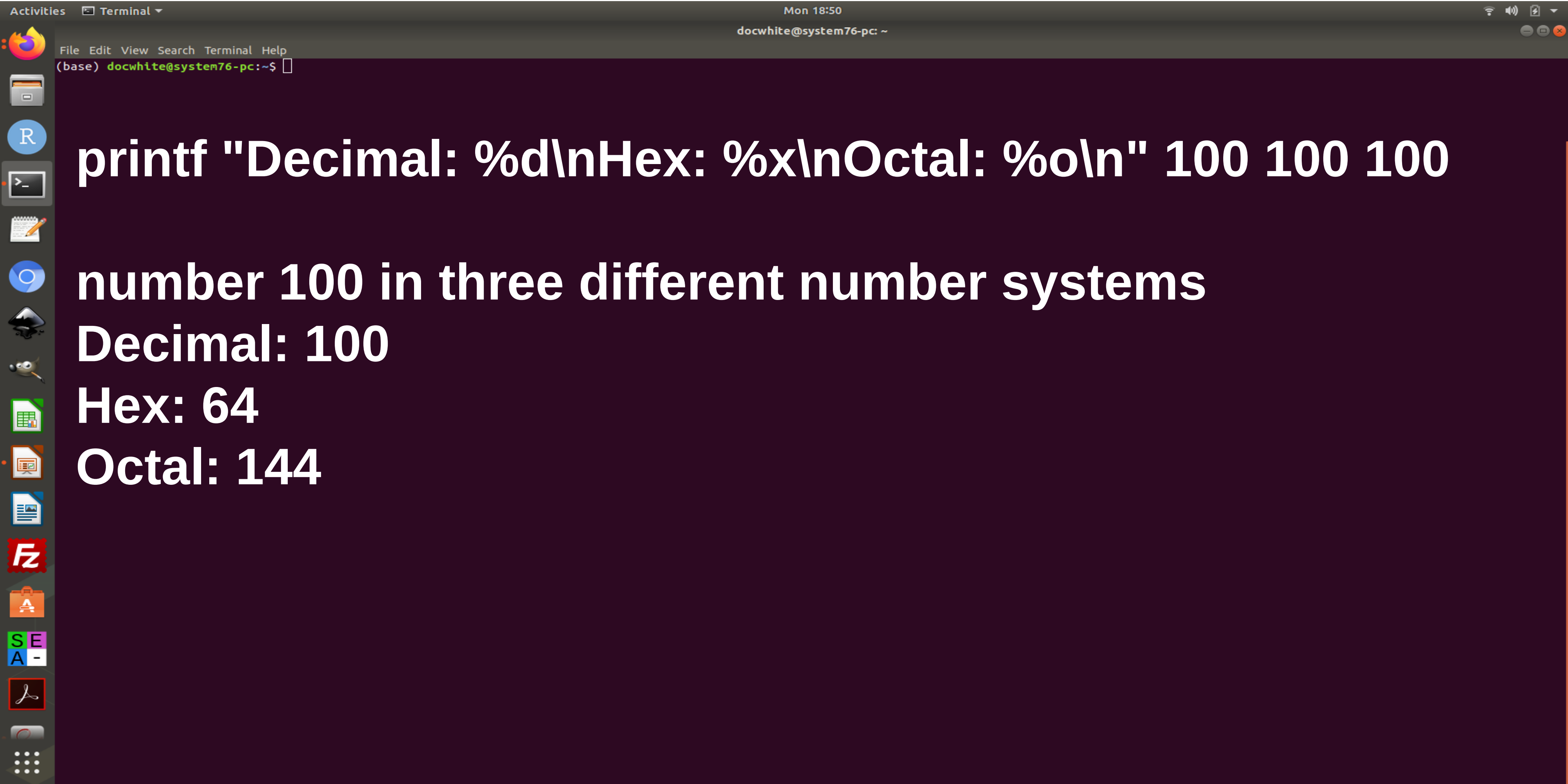
printf examples



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

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

printf examples



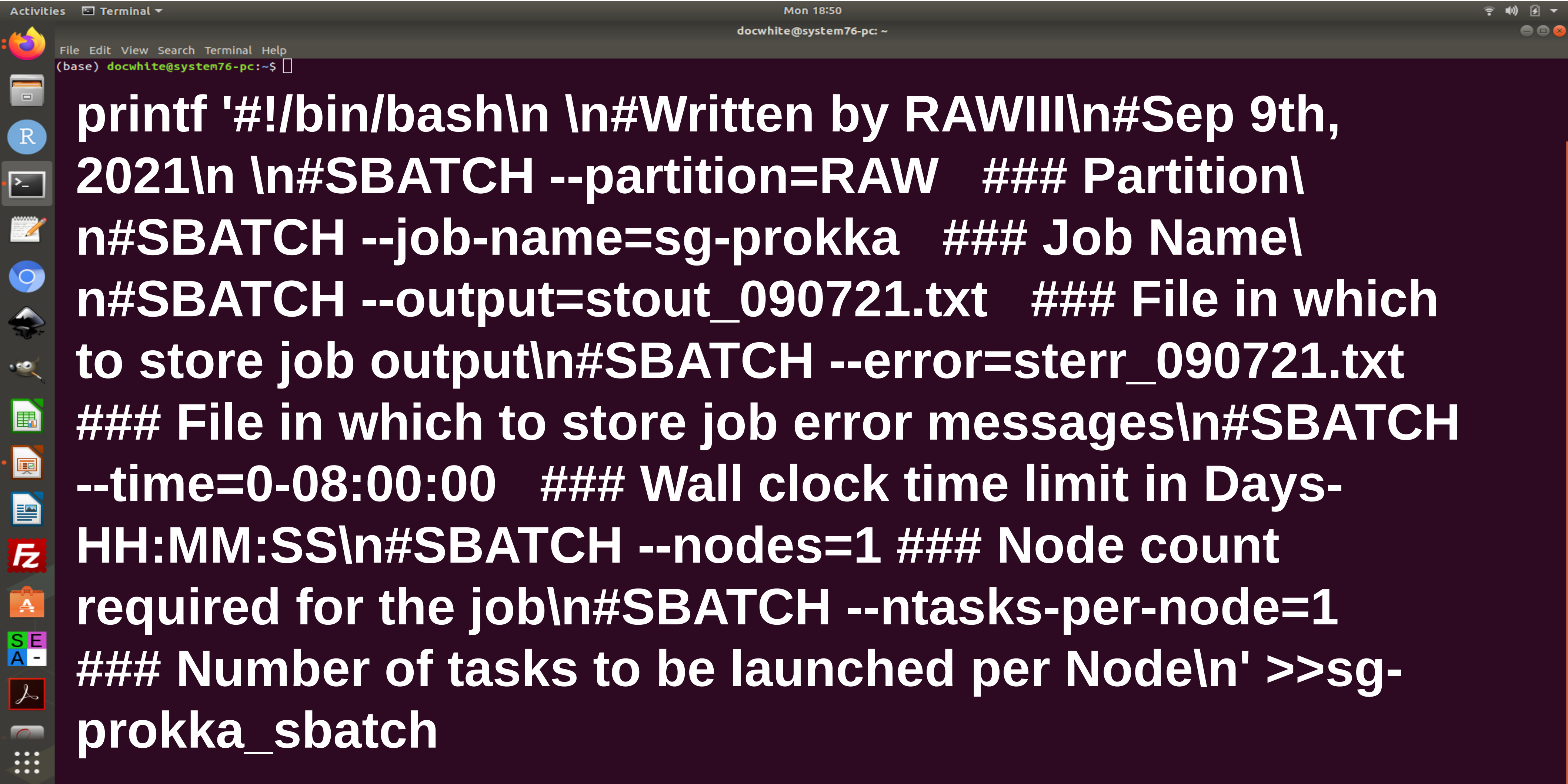
```
(base) docwhite@system76-pc:~$
```

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

number 100 in three different number systems

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

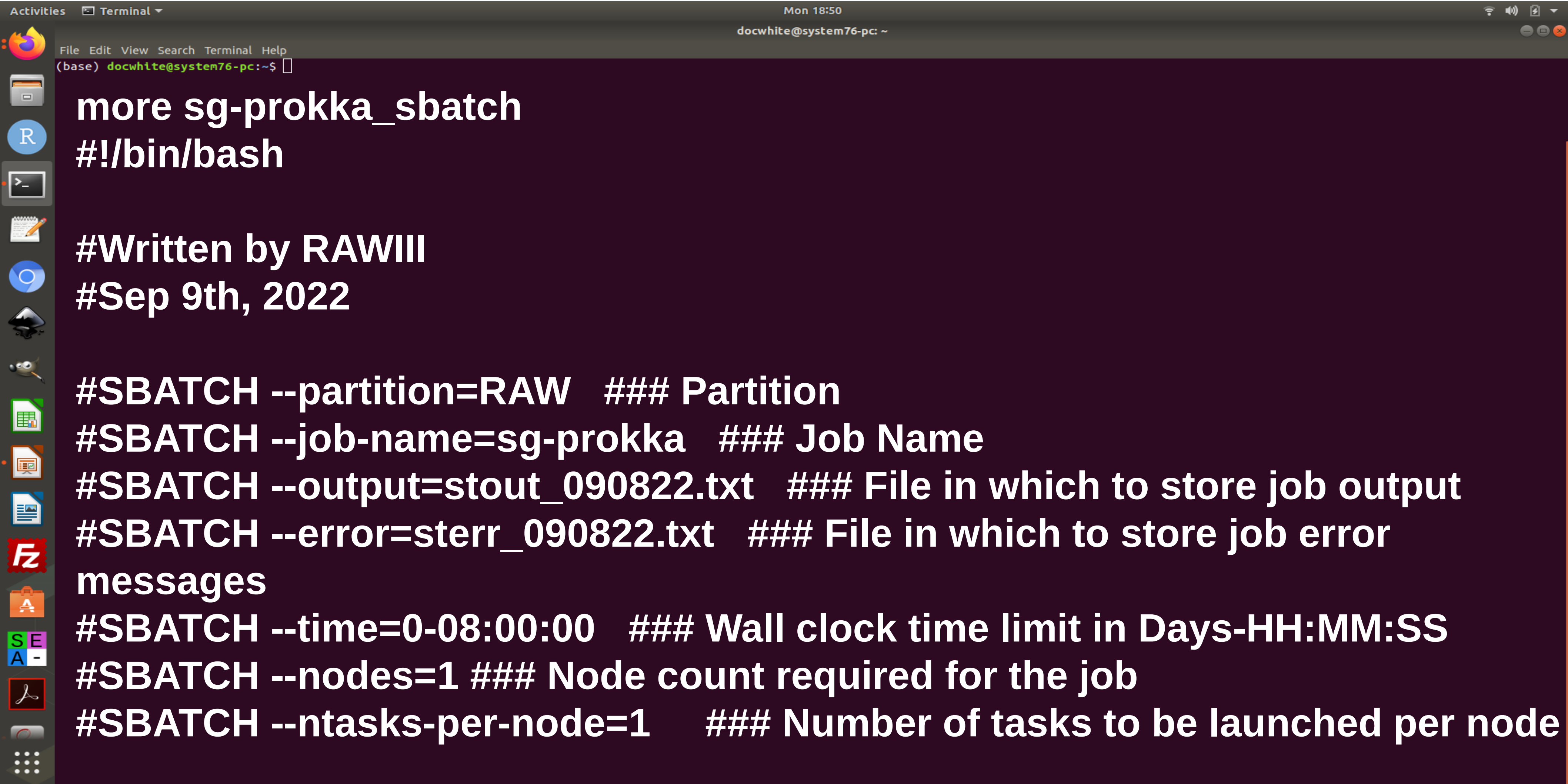
printf examples



```
(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
```

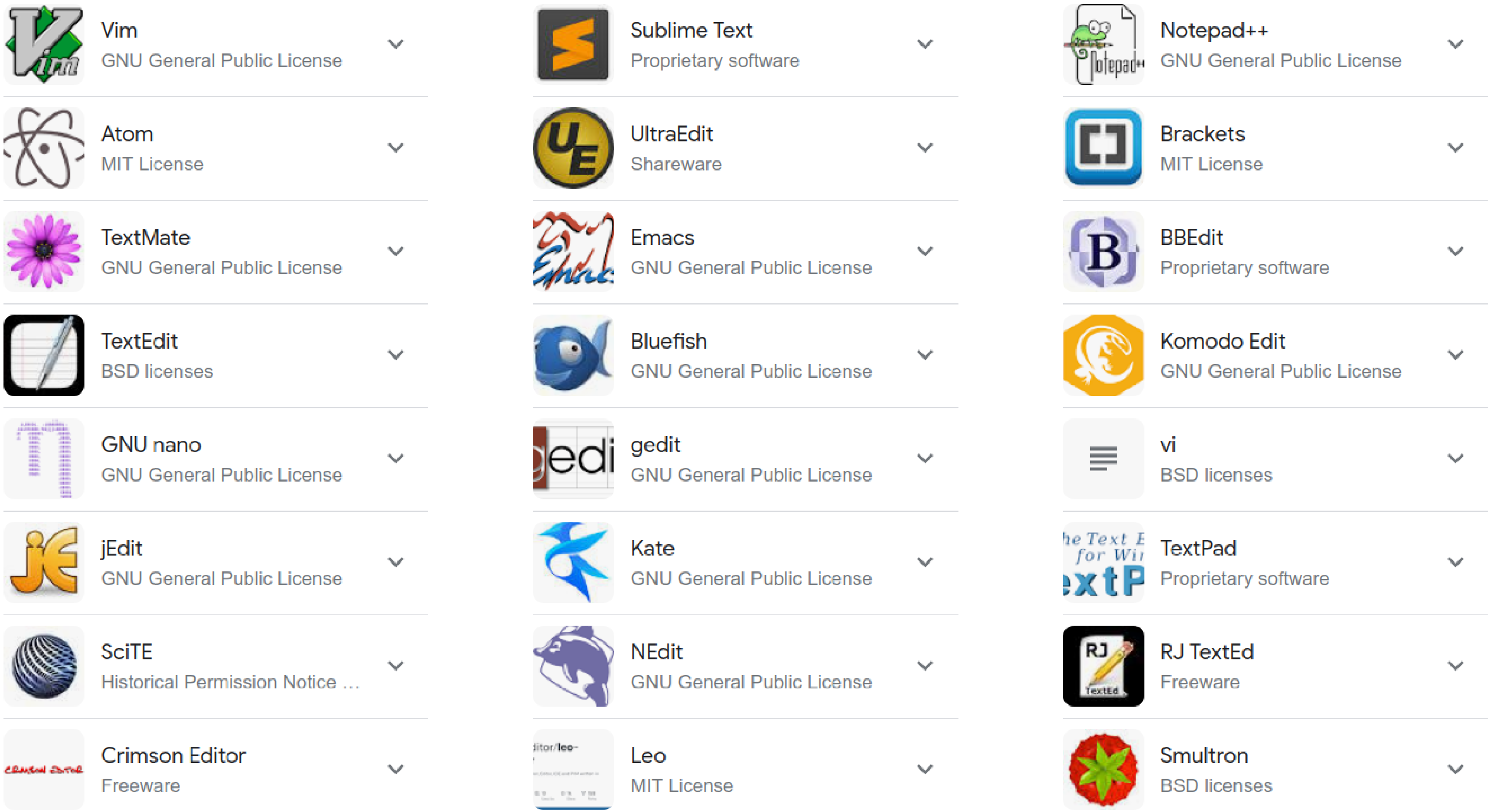
printf examples



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

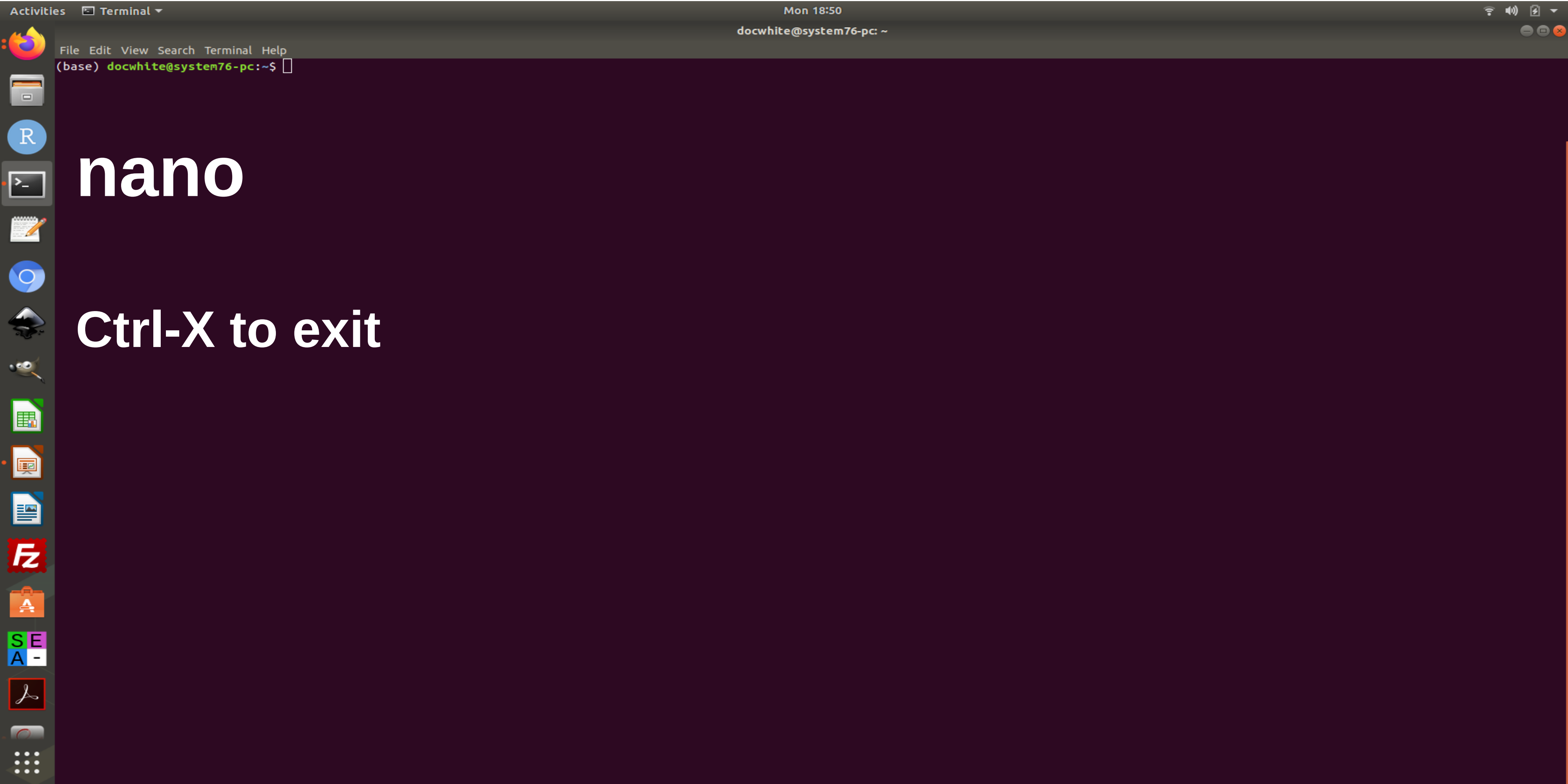
```
more sg-prokka_sbatch
#!/bin/bash

#Written by RAWIII
#Sep 9th, 2022

#SBATCH --partition=RAW   ### Partition
#SBATCH --job-name=sg-prokka   ### Job Name
#SBATCH --output=stout_090822.txt   ### File in which to store job output
#SBATCH --error=sterr_090822.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/IDE



Vim	GNU General Public License
Atom	MIT License
TextMate	GNU General Public License
TextEdit	BSD licenses
GNU nano	GNU General Public License
jEdit	GNU General Public License
SciTE	Historical Permission Notice …
Crimson Editor	Freeware
Sublime Text	Proprietary software
UltraEdit	Shareware
Emacs	GNU General Public License
Bluefish	GNU General Public License
gedit	GNU General Public License
Kate	GNU General Public License
NEdit	GNU General Public License
Leo	MIT License
Notepad++	GNU General Public License
Brackets	MIT License
BBEdit	Proprietary software
Komodo Edit	GNU General Public License
vi	BSD licenses
TextPad	Proprietary software
RJ TextEd	Freeware
Smultron	BSD licenses

Text editors - nano

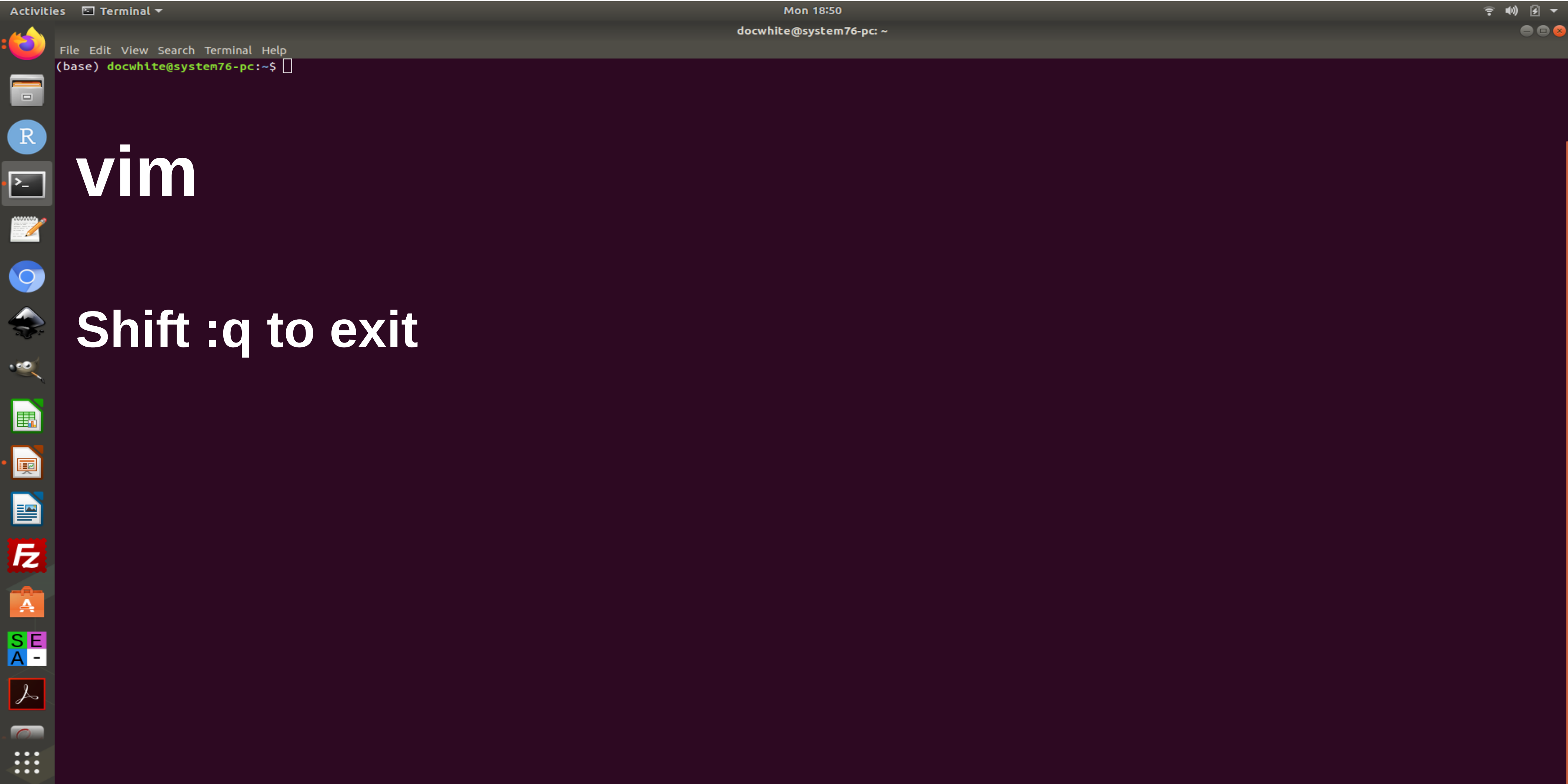


```
(base) docwhite@system76-pc:~$
```

nano

Ctrl-X to exit

Text editors - vim

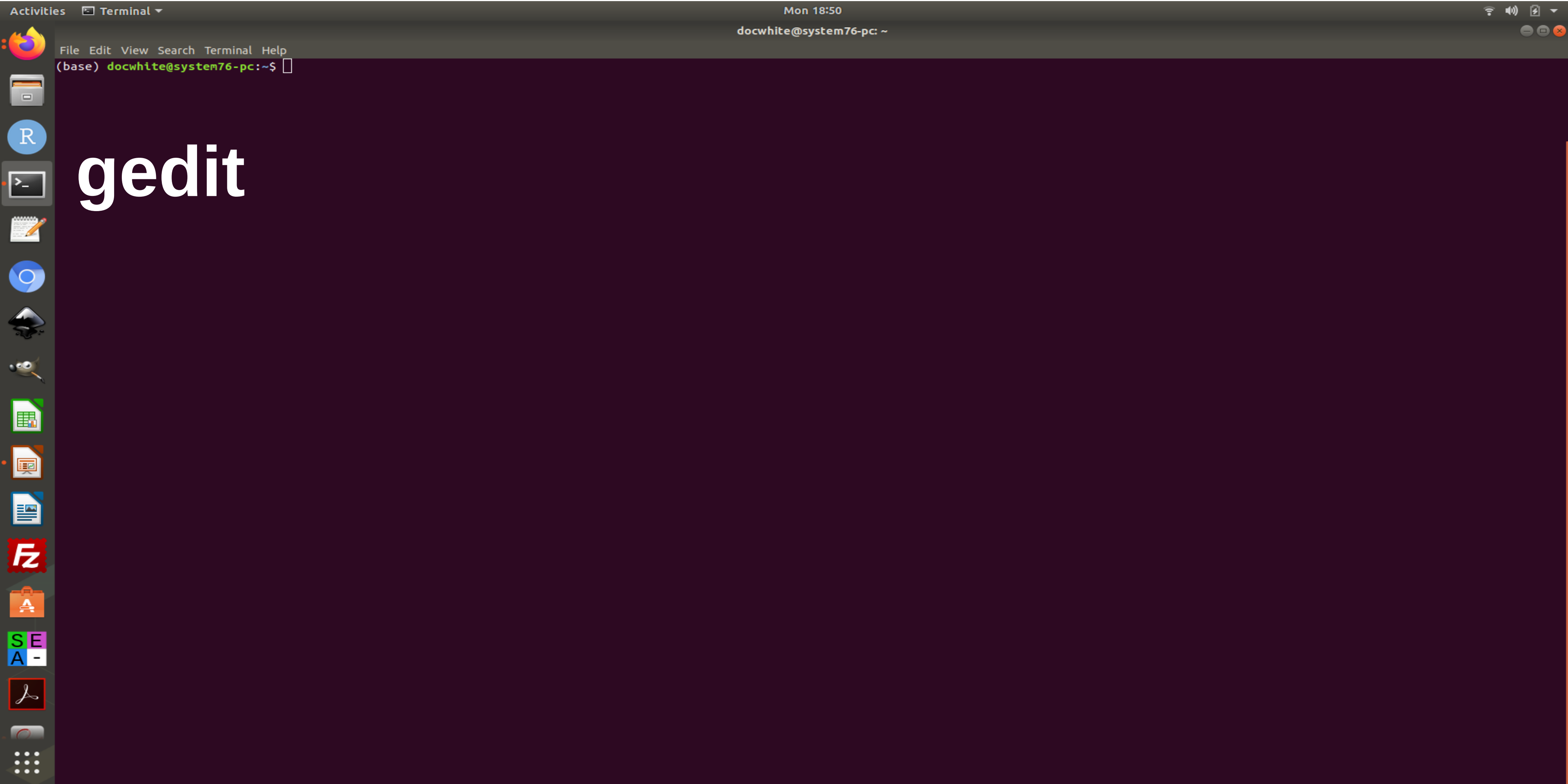


```
(base) docwhite@system76-pc:~$ 
```

vim

Shift :q to exit

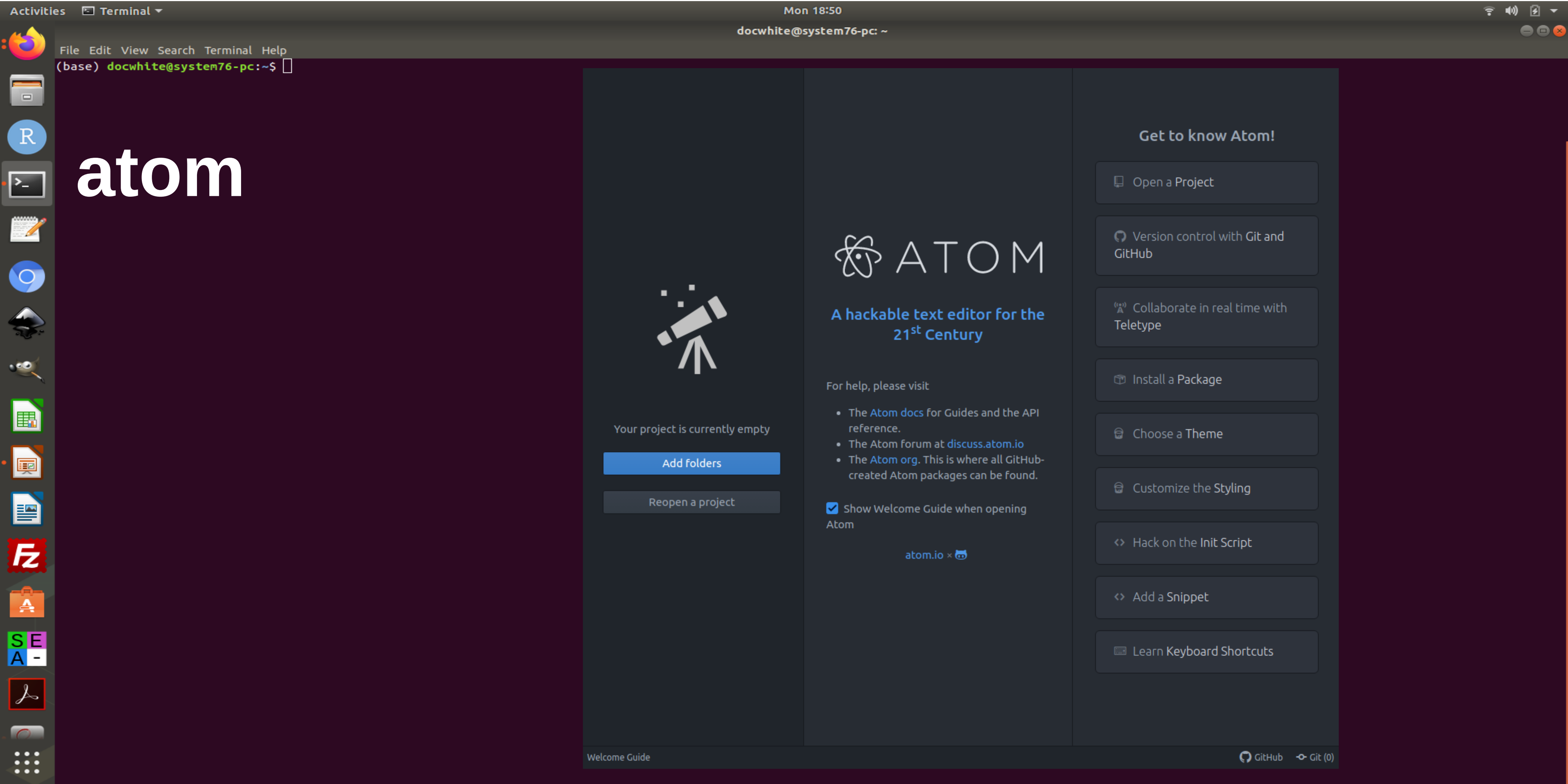
Text editors - gedit



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

gedit

Text editors - atom



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

atom

Your project is currently empty

ATOM

A hackable text editor for the 21st Century

For help, please visit

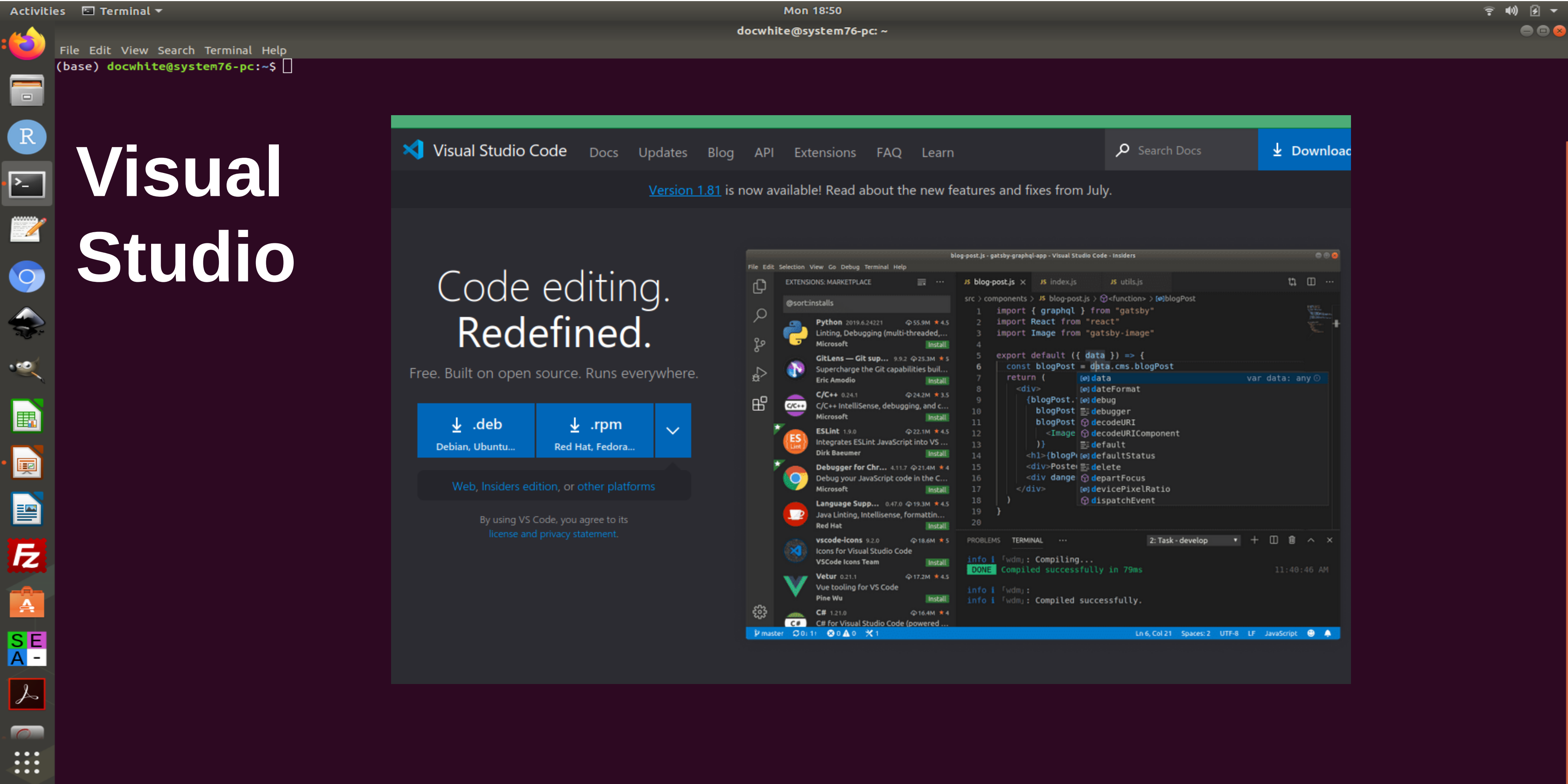
- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io
- The Atom org. This is where all GitHub-created Atom packages can be found.

☑ Show Welcome Guide when opening Atom

atom.io ×

Get to know Atom!

Text editors – Visual Studio



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

Visual Studio

Version 1.81 is now available! Read about the new features and fixes from July.

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

Web, Insiders edition, or other platforms

By using VS Code, you agree to its
license and privacy statement.

blog-post.js - gatsby-graphql-app - Visual Studio Code - Insiders
File Edit Selection View Go Debug Terminal Help
EXTENSIONS: MARKETPLACE
@sort:installs
blog-post.js
index.js
utils.js
src > components > blog-post.js > <function> > blogPost

Python 2019.6.24221 55.9M ★4.5 Linting, Debugging (multi-threaded,... Microsoft Install

GitLens — Git sup... 9.9.2 25.3M ★5 Supercharge the Git capabilities buil... Eric Amodio Install

C/C++ 0.24.1 24.2M ★3.5 C/C++ IntelliSense, debugging, and c... Microsoft Install

ESLint 1.9.0 22.1M ★4.5 Integrates ESLint JavaScript into VS ... Dirk Baeumer Install

Debugger for Chr... 4.11.7 21.4M ★4 Debug your JavaScript code in the C... Microsoft Install

Language Supp... 0.47.0 19.3M ★4.5 Java Linting, Intellisense, formattin... Red Hat Install

vscode-icons 9.2.0 18.6M ★5 Icons for Visual Studio Code VSCode Icons Team Install

Vetur 0.21.1 17.2M ★4.5 Vue tooling for VS Code Pine Wu Install

C# 1.21.0 16.4M ★4 C# for Visual Studio Code (powered ...

```
import { graphql } from "gatsby"
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = data.cms.blogPost
  return (
    <div>
      {blogPost.
        blogPost
        blogPost
          <Image
        )}
      <h1>{blogP
      <div>Poste
      <div dange
    </div>
  )
}
```

```
info ℹ ｢wdm｣: Compiling...
DONE  Compiled successfully in 79ms                    11:40:46 AM

info ℹ ｢wdm｣:
info ℹ ｢wdm｣: Compiled successfully.
```

Quiz 6

- On canvas now

Bonus 6

- 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: