Lecture 7

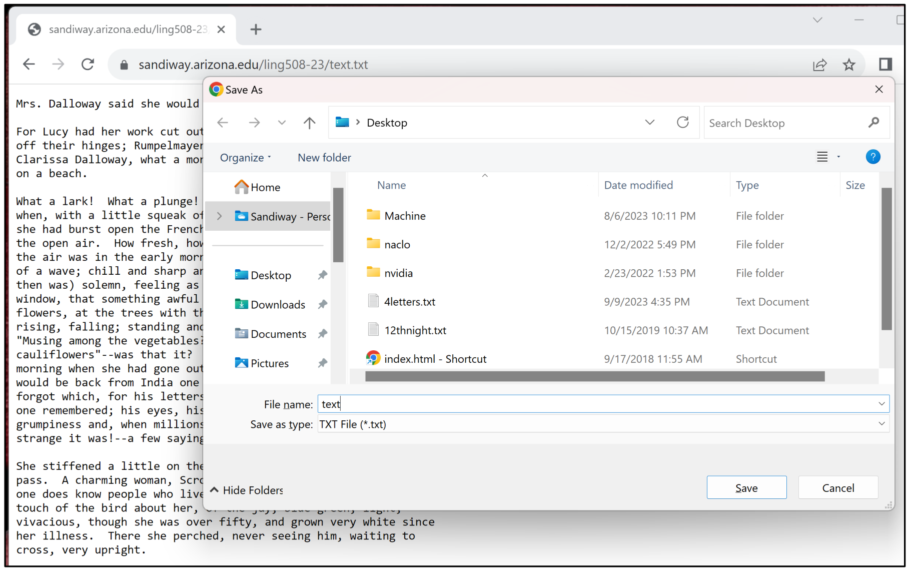
408/508 *Computational Techniques for Linguists*

Today's Topics

- Homework 3 review
  - *Step-by-step Bash shell exercises*
- Other things we can 'pipe' (|) into our workflow:
  - `tail`
  - `awk`
  - `termgraph`
- a note on file permissions

Windows into WSL2 (Ubuntu)

Lecture 4:
can access your Windows C: drive
(from within Ubuntu) via directory `/mnt/c`



sandiway.arizona.edu/ling508-23/text.txt

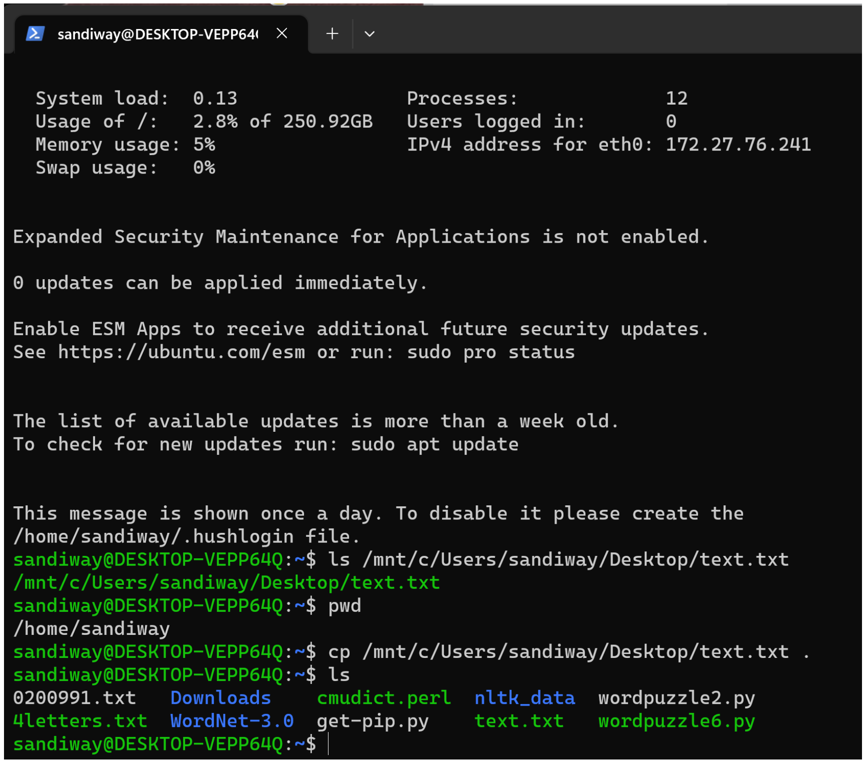
```
Mrs. Dalloway said she would

For Lucy had her work cut out
off their hinges; Rumpelmayer
Clarissa Dalloway, what a mor
on a beach.

What a lark!  What a plunge!
when, with a little squeak of
she had burst open the French
the open air.  How fresh, how
the air was in the early morn
of a wave; chill and sharp an
then was) solemn, feeling as
window, that something awful
flowers, at the trees with th
rising, falling; standing and
"Musing among the vegetables?
cauliflowers"--was that it?
morning when she had gone out
would be back from India one
forgot which, for his letters
one remembered; his eyes, his
grumpiness and, when millions
strange it was!--a few saying

She stiffened a little on the
pass.  A charming woman, Scro
one does know people who live
touch of the bird about her,
vivacious, though she was over fifty, and grown very white since
her illness.  There she perched, never seeing him, waiting to
cross, very upright.
```

Save As

Desktop

Name	Date modified	Type	Size
Machine	8/6/2023 10:11 PM	File folder	
naclo	12/2/2022 5:49 PM	File folder	
nvidia	2/23/2022 1:53 PM	File folder	
4letters.txt	9/9/2023 4:35 PM	Text Document	
12thnight.txt	10/15/2019 10:37 AM	Text Document	
index.html - Shortcut	9/17/2018 11:55 AM	Shortcut	

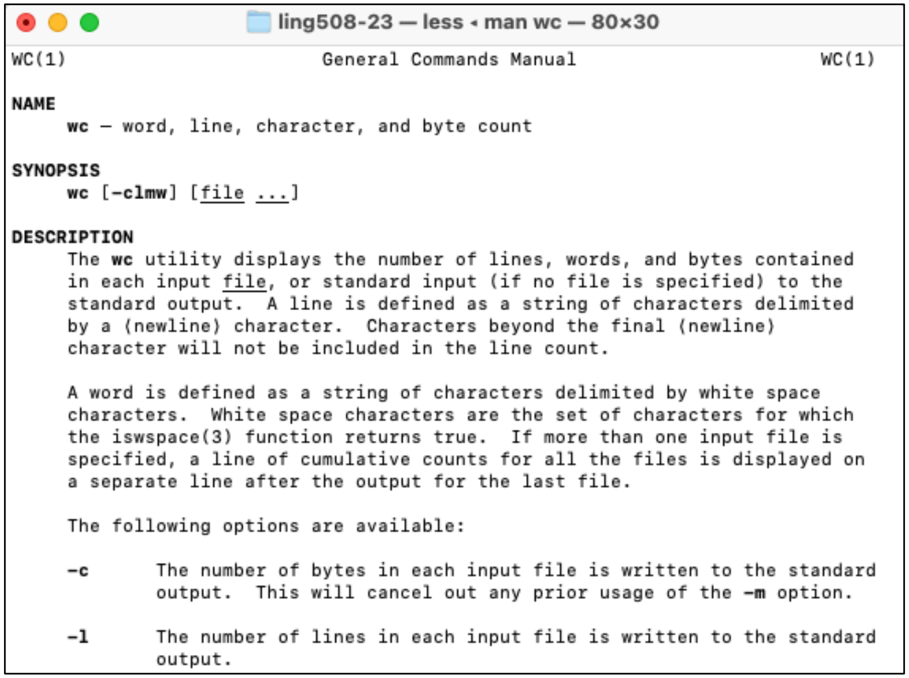
File name: text
Save as type: TXT File (*.txt)



```
  System load:  0.13                Processes:             12
  Usage of /:   2.8% of 250.92GB    Users logged in:       0
  Memory usage: 5%                  IPv4 address for eth0: 172.27.76.241
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


This message is shown once a day. To disable it please create the
/home/sandiway/.hushlogin file.
sandiway@DESKTOP-VEPP64Q:~$ ls /mnt/c/Users/sandiway/Desktop/text.txt
/mnt/c/Users/sandiway/Desktop/text.txt
sandiway@DESKTOP-VEPP64Q:~$ pwd
/home/sandiway
sandiway@DESKTOP-VEPP64Q:~$ cp /mnt/c/Users/sandiway/Desktop/text.txt .
sandiway@DESKTOP-VEPP64Q:~$ ls
0200991.txt  Downloads    cmudict.perl  nltk_data  wordpuzzle2.py
4letters.txt  WordNet-3.0  get-pip.py    text.txt   wordpuzzle6.py
sandiway@DESKTOP-VEPP64Q:~$
```

Homework 3: Exercise 1 Review

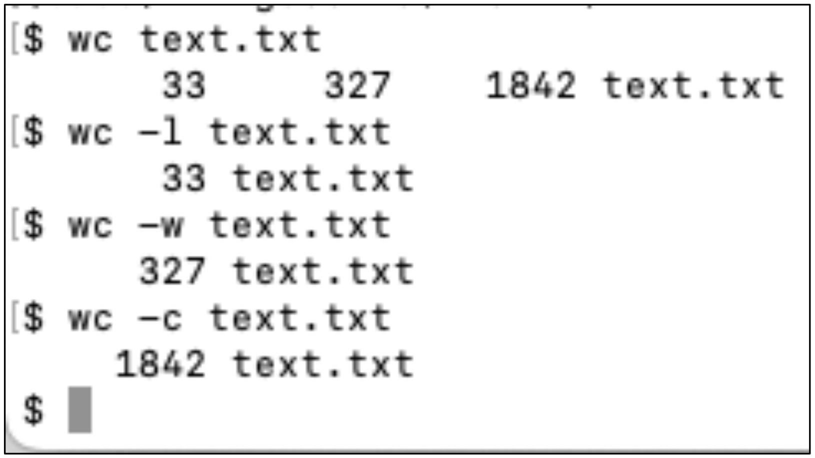
- Relevant bit:
  - `The **wc** utility displays the number of lines, words, and bytes`



```
WC(1)                        General Commands Manual                       WC(1)

NAME
     wc – word, line, character, and byte count

SYNOPSIS
     wc [-clmw] [file ...]

DESCRIPTION
     The wc utility displays the number of lines, words, and bytes contained
     in each input file, or standard input (if no file is specified) to the
     standard output.  A line is defined as a string of characters delimited
     by a ⟨newline⟩ character.  Characters beyond the final ⟨newline⟩
     character will not be included in the line count.

     A word is defined as a string of characters delimited by white space
     characters.  White space characters are the set of characters for which
     the iswspace(3) function returns true.  If more than one input file is
     specified, a line of cumulative counts for all the files is displayed on
     a separate line after the output for the last file.

     The following options are available:

     -c      The number of bytes in each input file is written to the standard
             output.  This will cancel out any prior usage of the -m option.

     -l      The number of lines in each input file is written to the standard
             output.
```

Homework 3: Exercise 1 Review

5. What's the wc option that prints the number of words only? Try it.



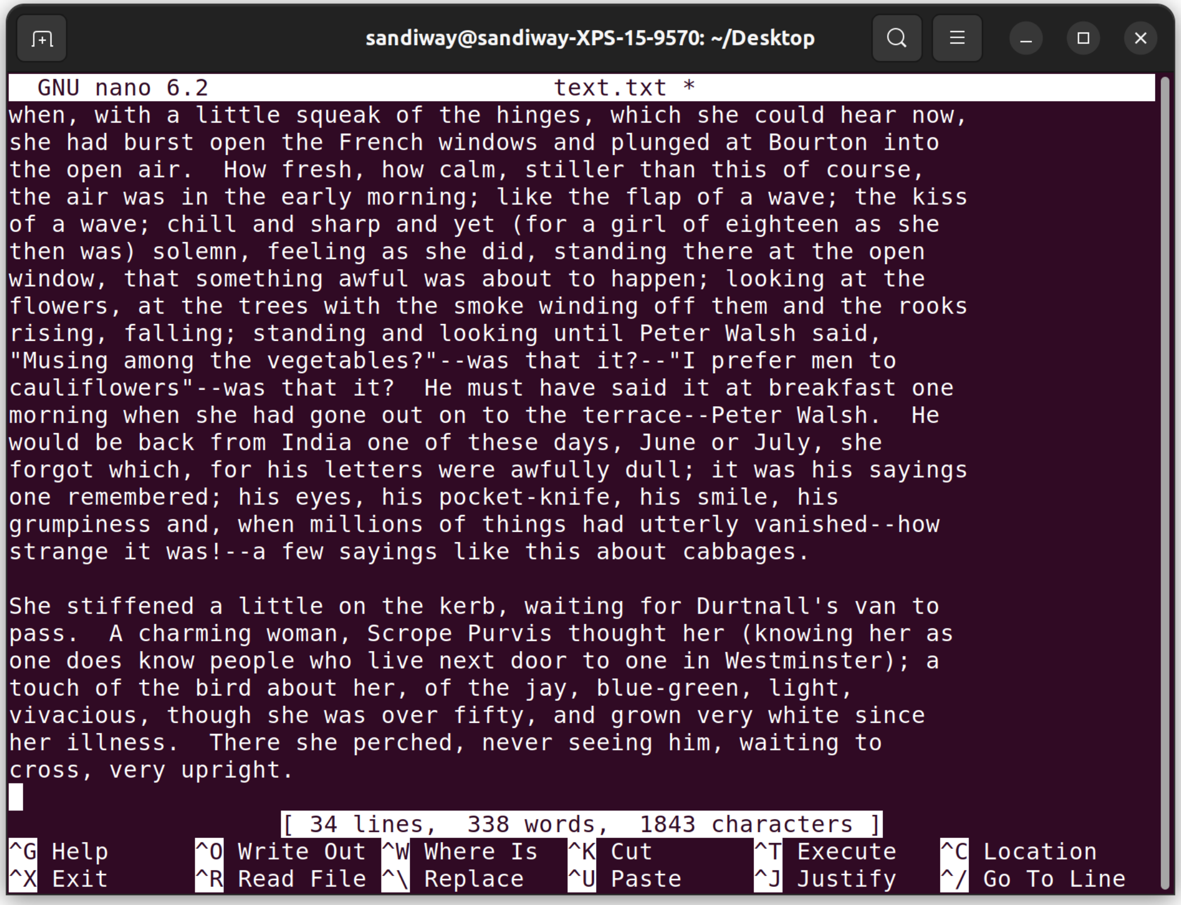
```
[$ wc text.txt
      33     327    1842 text.txt
[$ wc -l text.txt
      33 text.txt
[$ wc -w text.txt
     327 text.txt
[$ wc -c text.txt
    1842 text.txt
 $ 
```

Homework 3: Exercise 1 Review

`nano text.txt`

Type `Control-G`

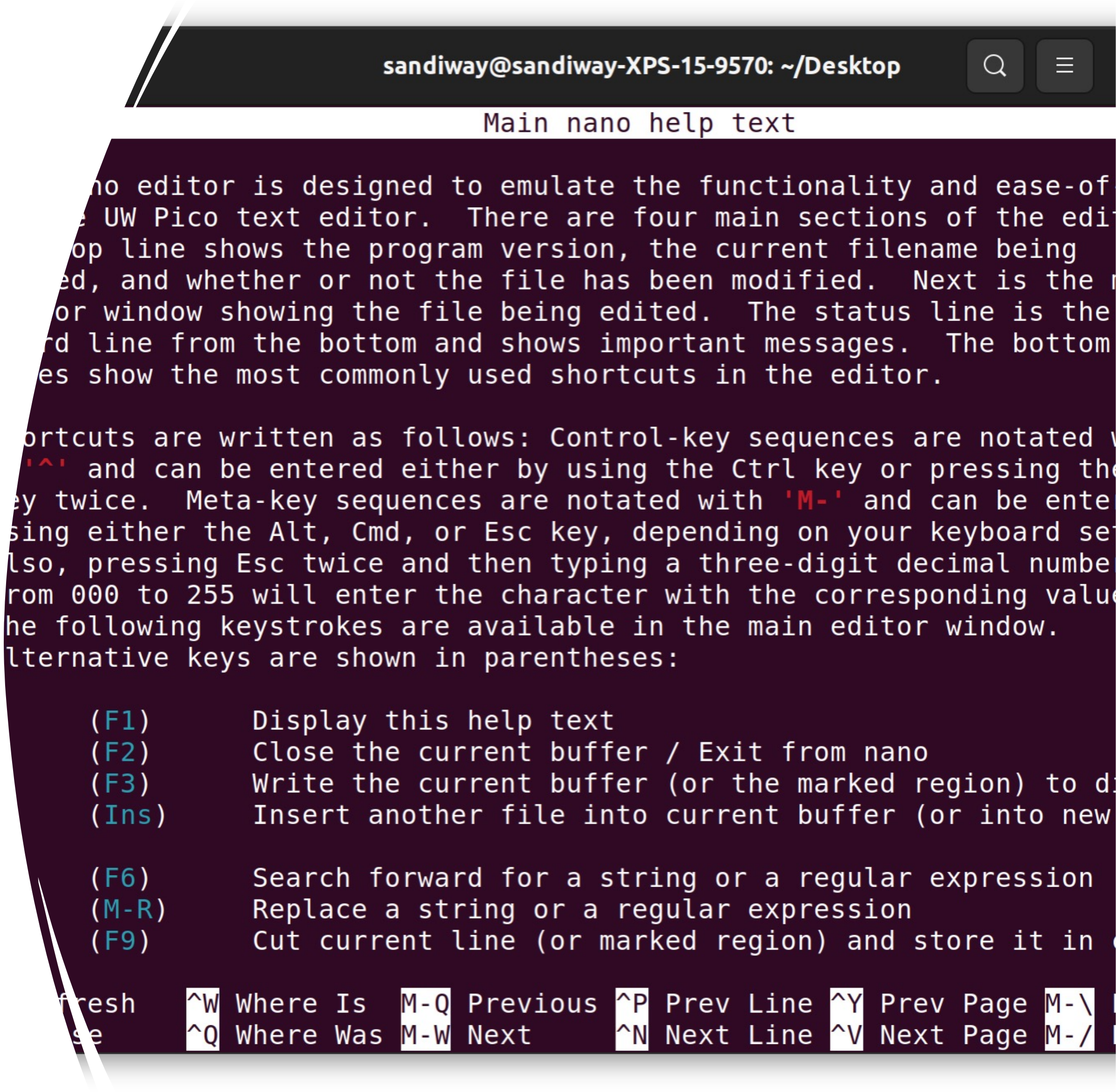
Type `Alt-D`



```
  GNU nano 6.2                                text.txt *
when, with a little squeak of the hinges, which she could hear now,
she had burst open the French windows and plunged at Bourton into
the open air.  How fresh, how calm, stiller than this of course,
the air was in the early morning; like the flap of a wave; the kiss
of a wave; chill and sharp and yet (for a girl of eighteen as she
then was) solemn, feeling as she did, standing there at the open
window, that something awful was about to happen; looking at the
flowers, at the trees with the smoke winding off them and the rooks
rising, falling; standing and looking until Peter Walsh said,
"Musing among the vegetables?"--was that it?--"I prefer men to
cauliflowers"--was that it?  He must have said it at breakfast one
morning when she had gone out on to the terrace--Peter Walsh.  He
would be back from India one of these days, June or July, she
forgot which, for his letters were awfully dull; it was his sayings
one remembered; his eyes, his pocket-knife, his smile, his
grumpiness and, when millions of things had utterly vanished--how
strange it was!--a few sayings like this about cabbages.

She stiffened a little on the kerb, waiting for Durtnall's van to
pass.  A charming woman, Scrope Purvis thought her (knowing her as
one does know people who live next door to one in Westminster); a
touch of the bird about her, of the jay, blue-green, light,
vivacious, though she was over fifty, and grown very white since
her illness.  There she perched, never seeing him, waiting to
cross, very upright.

                   [ 34 lines,  338 words,  1843 characters ]
^G Help         ^O Write Out ^W Where Is  ^K Cut        ^T Execute   ^C Location
^X Exit         ^R Read File ^\ Replace   ^U Paste      ^J Justify   ^/ Go To Line
```

Homework 3: Exercise 1 Review

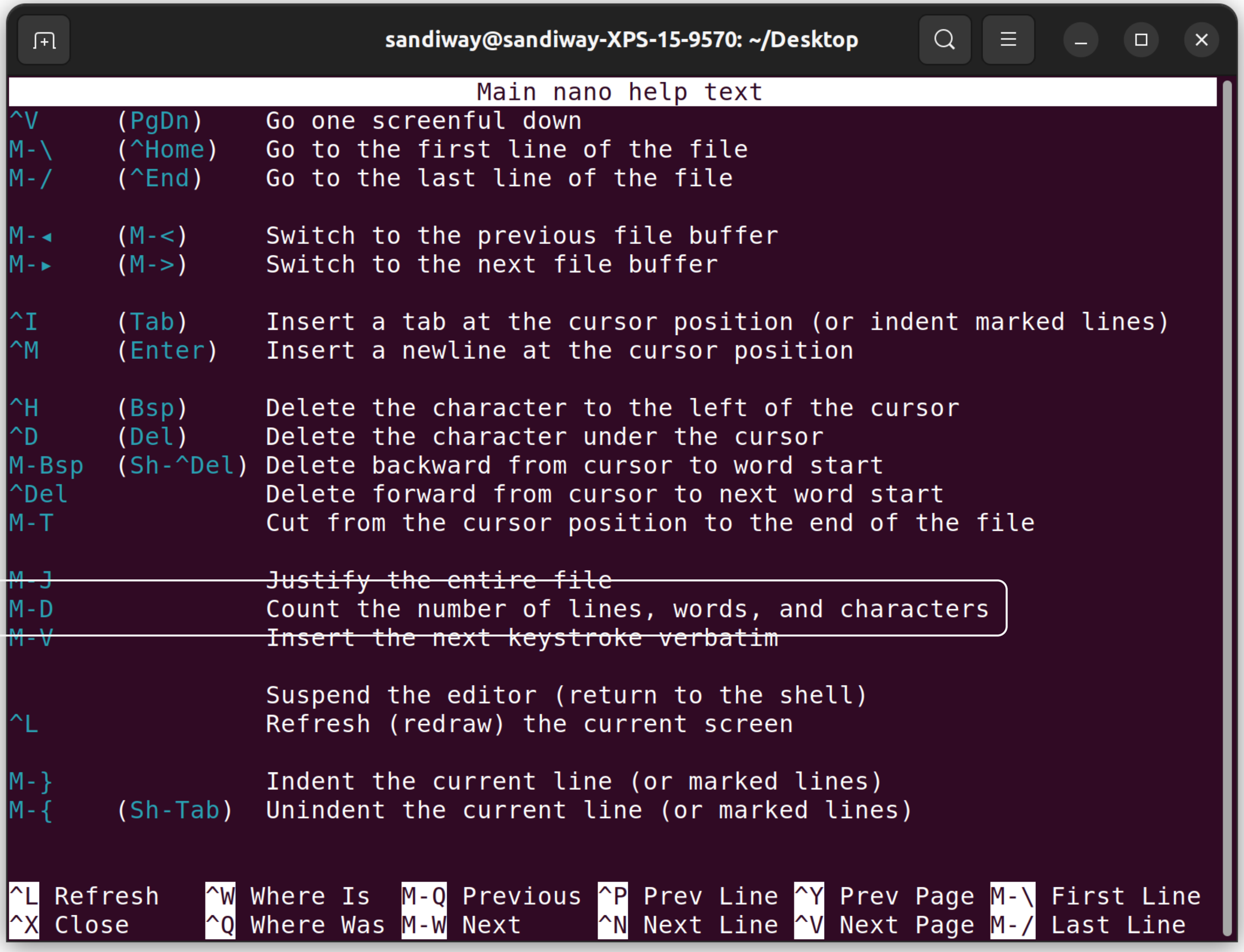
- Meta-key sequences are notated with M-
  - Alt, Cmd or Esc key



```
                  Main nano help text

no editor is designed to emulate the functionality and ease-of
 UW Pico text editor.  There are four main sections of the edi
op line shows the program version, the current filename being
ed, and whether or not the file has been modified.  Next is the m
or window showing the file being edited.  The status line is the
rd line from the bottom and shows important messages.  The bottom
es show the most commonly used shortcuts in the editor.

ortcuts are written as follows: Control-key sequences are notated w
'^' and can be entered either by using the Ctrl key or pressing the
ey twice.  Meta-key sequences are notated with 'M-' and can be ente
sing either the Alt, Cmd, or Esc key, depending on your keyboard se
lso, pressing Esc twice and then typing a three-digit decimal numbe
rom 000 to 255 will enter the character with the corresponding valu
he following keystrokes are available in the main editor window.
lternative keys are shown in parentheses:

    (F1)      Display this help text
    (F2)      Close the current buffer / Exit from nano
    (F3)      Write the current buffer (or the marked region) to di
    (Ins)     Insert another file into current buffer (or into new

    (F6)      Search forward for a string or a regular expression
    (M-R)     Replace a string or a regular expression
    (F9)      Cut current line (or marked region) and store it in c

fresh    ^W Where Is  M-Q Previous  ^P Prev Line  ^Y Prev Page  M-\
se       ^Q Where Was M-W Next      ^N Next Line  ^V Next Page  M-/
```

Homework 3: Exercise 1 Review

- M-D

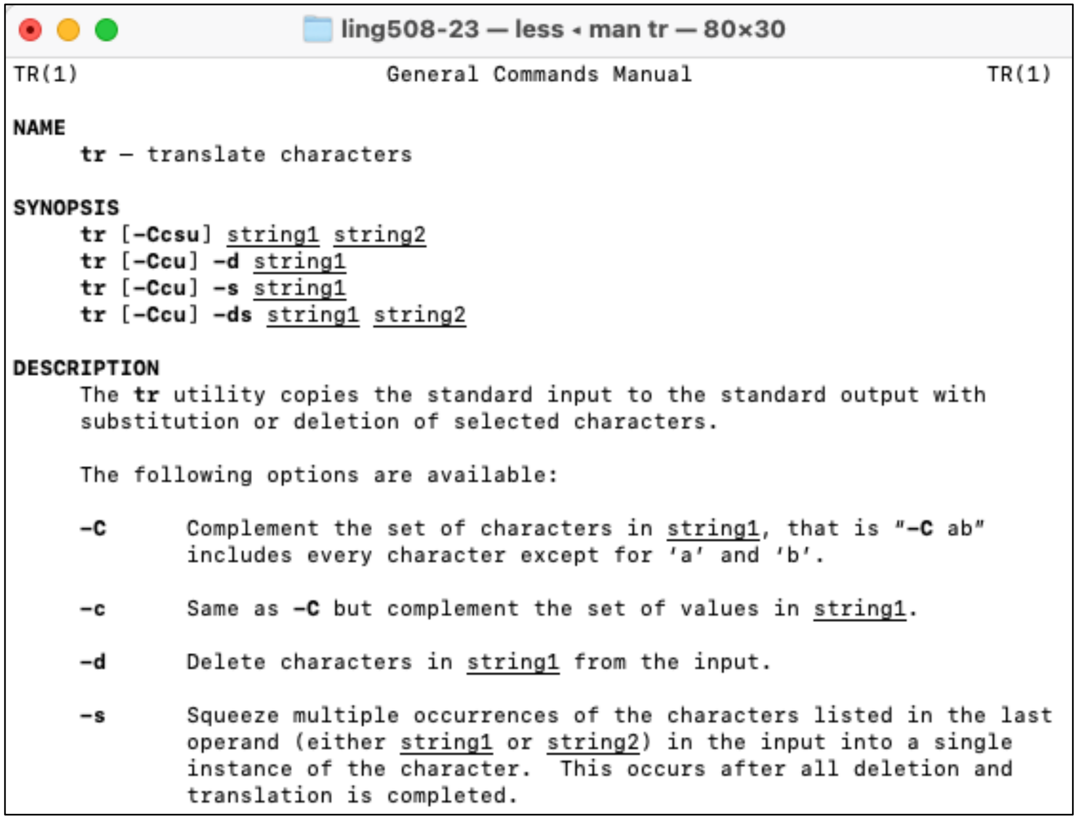


sandiway@sandiway-XPS-15-9570: ~/Desktop

```
                          Main nano help text
^V       (PgDn)    Go one screenful down
M-\      (^Home)   Go to the first line of the file
M-/      (^End)    Go to the last line of the file

M-◂      (M-<)     Switch to the previous file buffer
M-▸      (M->)     Switch to the next file buffer

^I       (Tab)     Insert a tab at the cursor position (or indent marked lines)
^M       (Enter)   Insert a newline at the cursor position

^H       (Bsp)     Delete the character to the left of the cursor
^D       (Del)     Delete the character under the cursor
M-Bsp    (Sh-^Del) Delete backward from cursor to word start
^Del               Delete forward from cursor to next word start
M-T                Cut from the cursor position to the end of the file

~~M-J                Justify the entire file~~
M-D                Count the number of lines, words, and characters
~~M-V                Insert the next keystroke verbatim~~

                   Suspend the editor (return to the shell)
^L                 Refresh (redraw) the current screen

M-}                Indent the current line (or marked lines)
M-{      (Sh-Tab)  Unindent the current line (or marked lines)


^L Refresh    ^W Where Is  M-Q Previous  ^P Prev Line  ^Y Prev Page  M-\ First Line
^X Close      ^Q Where Was M-W Next      ^N Next Line  ^V Next Page  M-/ Last Line
```

Homework 3: Exercise 2 Review

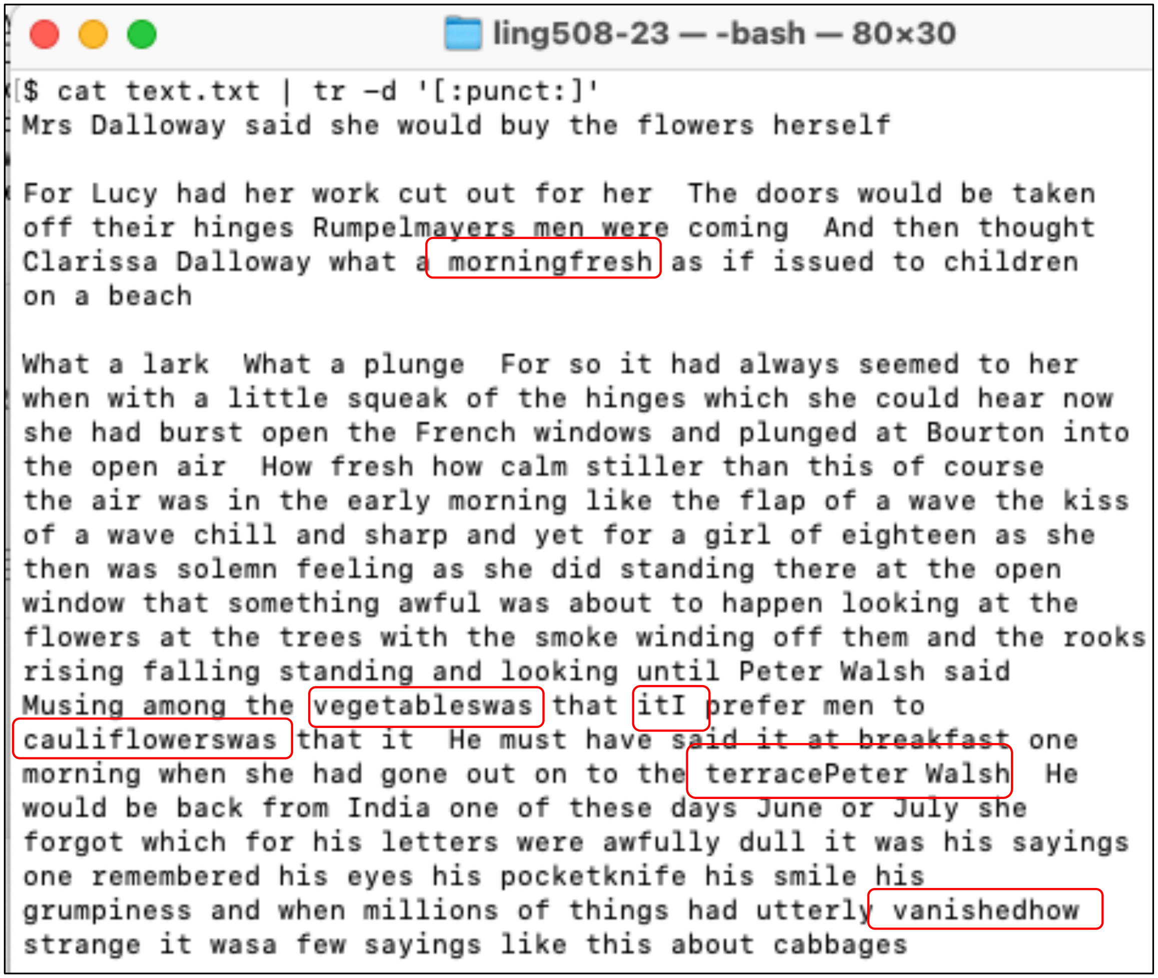
- Let's use the Terminal to make a frequency list of the words in `text.txt`
- First, look at the manpage for command `tr`.



```
TR(1)                      General Commands Manual                     TR(1)

NAME
     tr – translate characters

SYNOPSIS
     tr [-Ccsu] string1 string2
     tr [-Ccu] -d string1
     tr [-Ccu] -s string1
     tr [-Ccu] -ds string1 string2

DESCRIPTION
     The tr utility copies the standard input to the standard output with
     substitution or deletion of selected characters.

     The following options are available:

     -C      Complement the set of characters in string1, that is "-C ab"
             includes every character except for 'a' and 'b'.

     -c      Same as -C but complement the set of values in string1.

     -d      Delete characters in string1 from the input.

     -s      Squeeze multiple occurrences of the characters listed in the last
             operand (either string1 or string2) in the input into a single
             instance of the character.  This occurs after all deletion and
             translation is completed.
```

Homework 3: Exercise 2 Review

- First, look at the manpage for command tr.
- Next, let's replace all the punctuation characters by spaces.
- Observe the output of (either):
- `cat text.txt | tr '[:punct:]' ' '`
- `cat text.txt | tr -d '[:punct:]'`

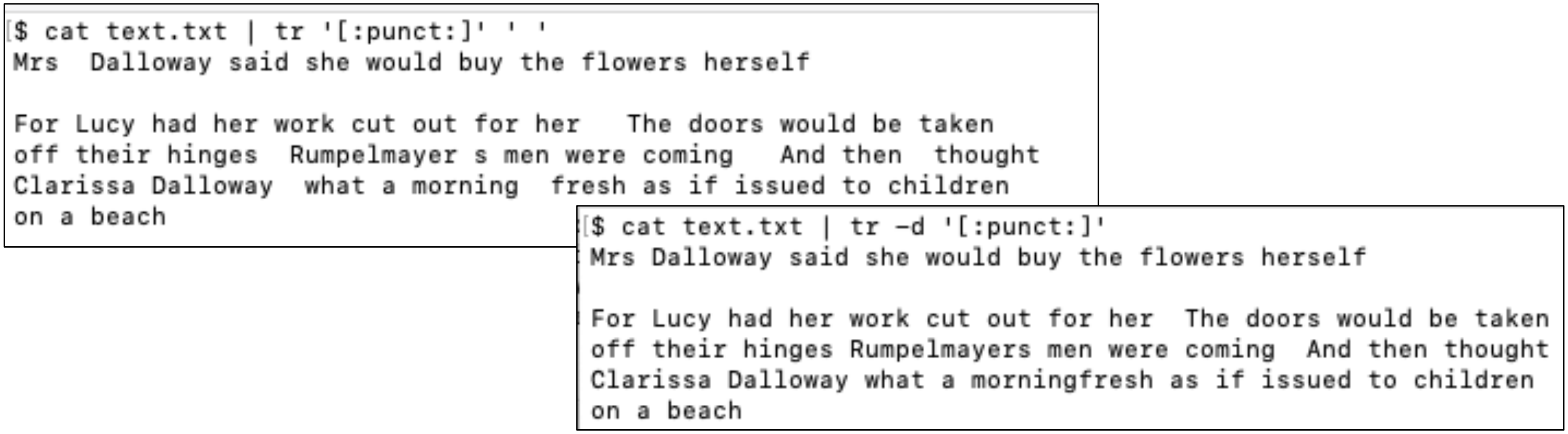


```
[$ cat text.txt | tr -d '[:punct:]'
Mrs Dalloway said she would buy the flowers herself

For Lucy had her work cut out for her  The doors would be taken
off their hinges Rumpelmayers men were coming  And then thought
Clarissa Dalloway what a morningfresh as if issued to children
on a beach

What a lark  What a plunge  For so it had always seemed to her
when with a little squeak of the hinges which she could hear now
she had burst open the French windows and plunged at Bourton into
the open air  How fresh how calm stiller than this of course
the air was in the early morning like the flap of a wave the kiss
of a wave chill and sharp and yet for a girl of eighteen as she
then was solemn feeling as she did standing there at the open
window that something awful was about to happen looking at the
flowers at the trees with the smoke winding off them and the rooks
rising falling standing and looking until Peter Walsh said
Musing among the vegetableswas that itI prefer men to
cauliflowerswas that it  He must have said it at breakfast one
morning when she had gone out on to the terracePeter Walsh  He
would be back from India one of these days June or July she
forgot which for his letters were awfully dull it was his sayings
one remembered his eyes his pocketknife his smile his
grumpiness and when millions of things had utterly vanishedhow
strange it wasa few sayings like this about cabbages
```

Homework 3: Exercise 2 review

- Next, let's replace all the punctuation characters by spaces.

1. Observe the output of both commands below. Which command do we want?
    - `cat text.txt | tr '[:punct:]' ' '`
    - `cat text.txt | tr –d '[:punct:]'`



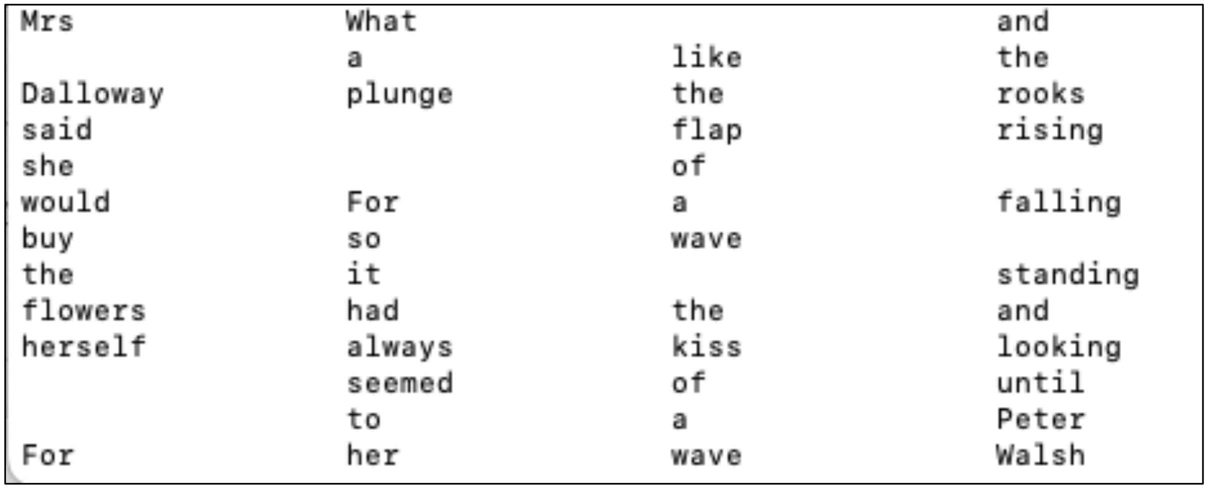
```
$ cat text.txt | tr '[:punct:]' ' '
Mrs  Dalloway said she would buy the flowers herself

For Lucy had her work cut out for her   The doors would be taken
off their hinges  Rumpelmayer s men were coming   And then  thought
Clarissa Dalloway  what a morning  fresh as if issued to children
on a beach
```

```
$ cat text.txt | tr –d '[:punct:]'
Mrs Dalloway said she would buy the flowers herself

For Lucy had her work cut out for her  The doors would be taken
off their hinges Rumpelmayers men were coming  And then thought
Clarissa Dalloway what a morningfresh as if issued to children
on a beach
```

Homework 3: Exercise 2 Review

2. Next, we can put each word on a separate line using:
   - `tr ' ' '\n'`
   - **Note 3**: \n stands for a newline character.

```
$ cat text.txt | tr '[:punct:]' ' ' | tr ' ' '\n' | pr –t4
```

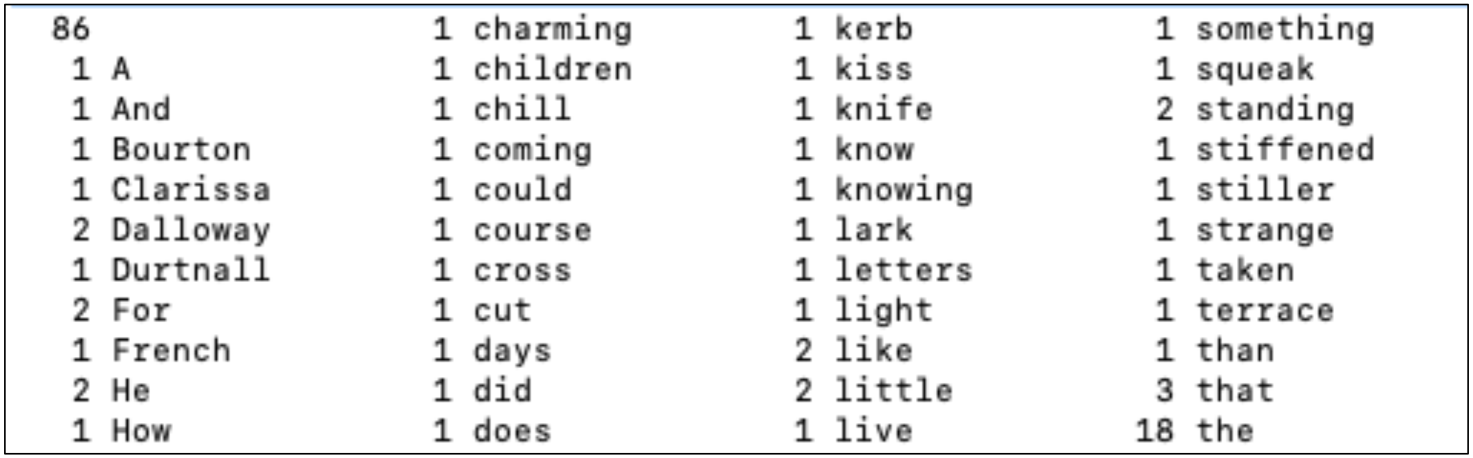


```
Mrs               What                                  and
                  a                 like                the
Dalloway          plunge            the                 rooks
said                                flap                rising
she                                 of
would             For               a                   falling
buy               so                wave
the               it                                    standing
flowers           had               the                 and
herself           always            kiss                looking
                  seemed            of                  until
                  to                a                   Peter
For               her               wave                Walsh
```

Homework 3: Exercise 2 Review

4. Let's make a table of the frequency counts for each word using:
   - `sort | uniq –c`

```
cat text.txt | tr '[:punct:]' ' ' | tr ' ' '\n' | sort | uniq –c | pr –t4
```



```
  86                    1 charming          1 kerb              1 something
   1 A                  1 children          1 kiss              1 squeak
   1 And                1 chill             1 knife             2 standing
   1 Bourton            1 coming            1 know              1 stiffened
   1 Clarissa           1 could             1 knowing           1 stiller
   2 Dalloway           1 course            1 lark              1 strange
   1 Durtnall           1 cross             1 letters           1 taken
   2 For                1 cut               1 light             1 terrace
   1 French             1 days              2 like              1 than
   2 He                 1 did               2 little            3 that
   1 How                1 does              1 live             18 the
```

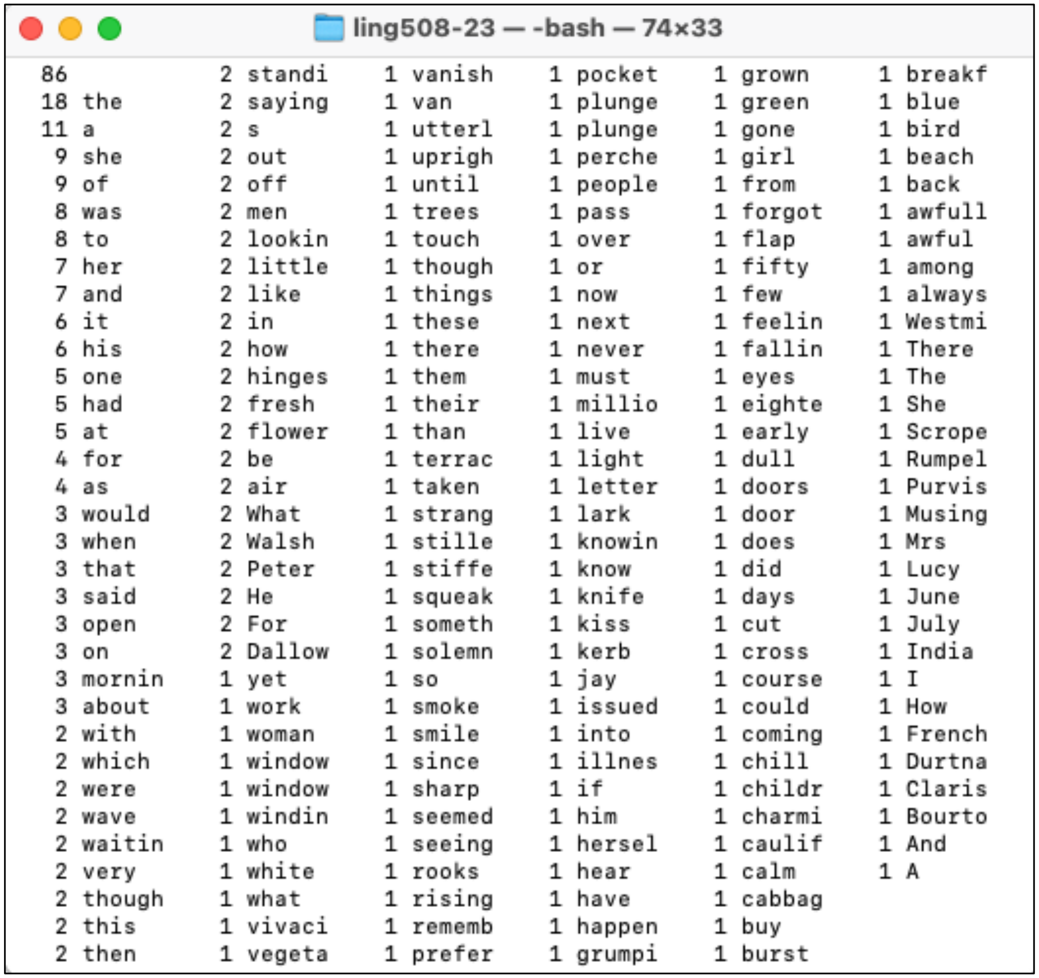
Recall ASCII table: A-Z comes before a-z.

Homework 3: Exercise 2

```
NAME
     uniq – report or filter out repeated lines in a file

SYNOPSIS
     uniq [-c | -d | -D | -u] [-i] [-f num] [-s chars] [input file [output file]]

DESCRIPTION
     The uniq utility reads the specified input file comparing adjacent lines, and writes a
     copy of each unique input line to the output file.
     The second and succeeding copies of identical adjacent input lines are not written.
     Repeated lines in the input will not be detected if they are not adjacent, so it may be
     necessary to sort the files first.
     The following options are available:
     -c, --count
             Precede each output line with the count of the number of times the line
             occurred in the input, followed by a single space.
```

Homework 3: Exercise 2 Review

6. Let's put the results in sorted order of frequency (*descending*) by appending:
   - `sort -rn`



```
  86            2 standi    1 vanish    1 pocket    1 grown     1 breakf
  18 the        2 saying    1 van       1 plunge    1 green     1 blue
  11 a          2 s         1 utterl    1 plunge    1 gone      1 bird
   9 she        2 out       1 uprigh    1 perche    1 girl      1 beach
   9 of         2 off       1 until     1 people    1 from      1 back
   8 was        2 men       1 trees     1 pass      1 forgot    1 awfull
   8 to         2 lookin    1 touch     1 over      1 flap      1 awful
   7 her        2 little    1 though    1 or        1 fifty     1 among
   7 and        2 like      1 things    1 now       1 few       1 always
   6 it         2 in        1 these     1 next      1 feelin    1 Westmi
   6 his        2 how       1 there     1 never     1 fallin    1 There
   5 one        2 hinges    1 them      1 must      1 eyes      1 The
   5 had        2 fresh     1 their     1 millio    1 eighte    1 She
   5 at         2 flower    1 than      1 live      1 early     1 Scrope
   4 for        2 be        1 terrac    1 light     1 dull      1 Rumpel
   4 as         2 air       1 taken     1 letter    1 doors     1 Purvis
   3 would      2 What      1 strang    1 lark      1 door      1 Musing
   3 when       2 Walsh     1 stille    1 knowin    1 does      1 Mrs
   3 that       2 Peter     1 stiffe    1 know      1 did       1 Lucy
   3 said       2 He        1 squeak    1 knife     1 days      1 June
   3 open       2 For       1 someth    1 kiss      1 cut       1 July
   3 on         2 Dallow    1 solemn    1 kerb      1 cross     1 India
   3 mornin     1 yet       1 so        1 jay       1 course    1 I
   3 about      1 work      1 smoke     1 issued    1 could     1 How
   2 with       1 woman     1 smile     1 into      1 coming    1 French
   2 which      1 window    1 since     1 illnes    1 chill     1 Durtna
   2 were       1 window    1 sharp     1 if        1 childr    1 Claris
   2 wave       1 windin    1 seemed    1 him       1 charmi    1 Bourto
   2 waitin     1 who       1 seeing    1 hersel    1 caulif    1 And
   2 very       1 white     1 rooks     1 hear      1 calm      1 A
   2 though     1 what      1 rising    1 have      1 cabbag
   2 this       1 vivaci    1 rememb    1 happen    1 buy
   2 then       1 vegeta    1 prefer    1 grumpi    1 burst
```

Homework 3: Exercise 2 Review

```
NAME
     sort – sort or merge records (lines) of text and binary files

SYNOPSIS
     sort [-bcCdfghiRMmnrsuVz] [-k field1[,field2]] [-S memsize] [-T dir] [-t char] [-o
          output] [file ...]

DESCRIPTION
     The sort utility sorts text and binary files by lines.

     -n, --numeric-sort, --sort=numeric
             Sort fields numerically by arithmetic value.  Fields are supposed to have
             optional blanks in the beginning, an optional minus sign, zero or more
             digits (including decimal point and possible thousand separators).
     -r, --reverse
             Sort in reverse order.
```

A step beyond Homework 3

Let's graph our homework result!

- There's something called `termgraph` (written in Python) but you can use it on the command line
- Assume you have python3 already installed
- Check whether it's already installed
  - `which termgraph`
  - `/Users/sandiway/opt/anaconda3/bin/termgraph`
- if not:
  - `pip3 install termgraph`

termgraph install

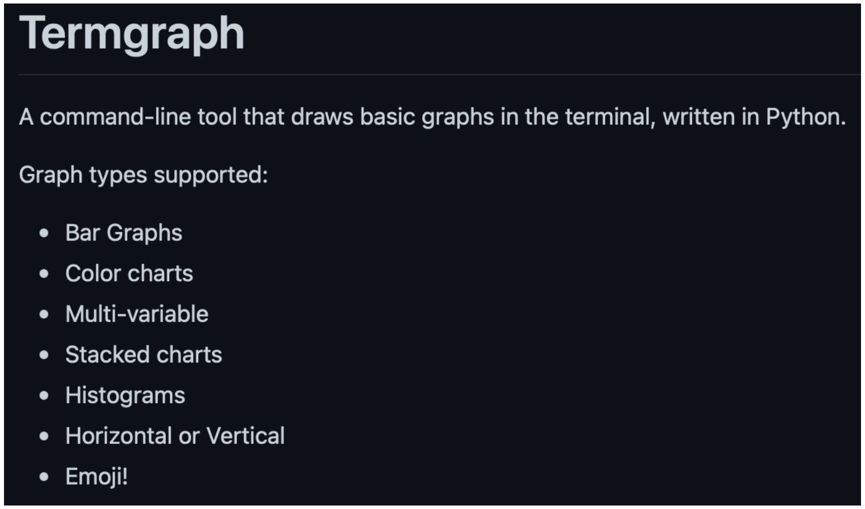
```
$ which termgraph  (no response means can't find the command)
$ pip3 install termgraph
Collecting termgraph
  Downloading termgraph-0.5.3-py3-none-any.whl (15 kB)
Collecting colorama
  Downloading colorama-0.4.5-py2.py3-none-any.whl (16 kB)
Installing collected packages: colorama, termgraph
Successfully installed colorama-0.4.5 termgraph-0.5.3
$ which termgraph
/opt/miniconda3/bin/termgraph
```

termgraph install

- It may place the executable in a directory that's not in your PATH.
- If so:
  - `export PATH=/home/yourname/.login/bin:$PATH`
  - will prepend `/home/yourname/.login/bin` to your PATH
  - and which termgraph should now work
- To make the change permanent, you can add this line to your startup file, either .bashrc or .bash_profile in your home directory (depending on which one exists)
  - `cd` (*goto home*)
  - `nano .bashrc` (*save change and exit*)

termgraph

- Google termgraph
  - https://github.com/mkaz/termgraph

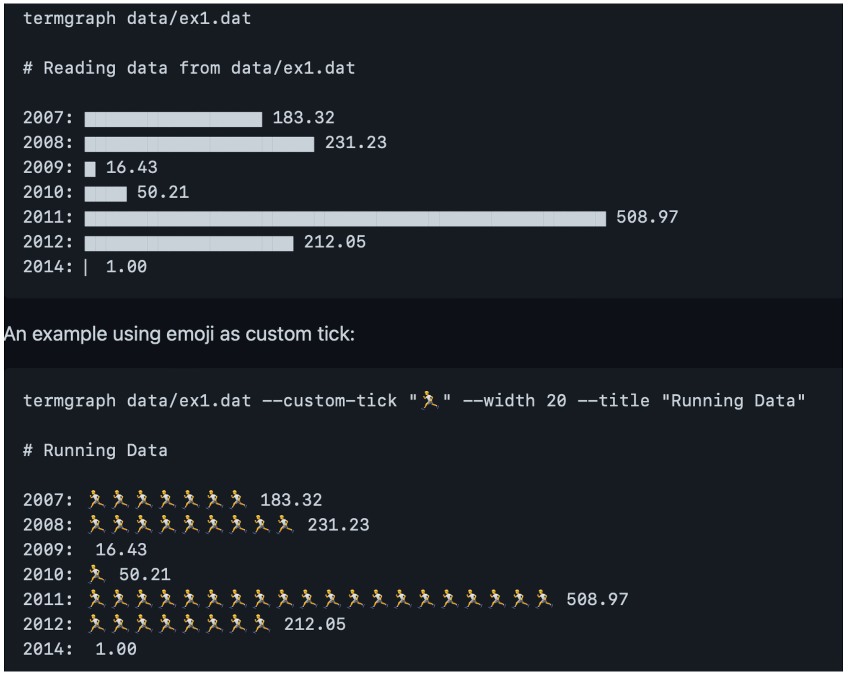


Termgraph

A command-line tool that draws basic graphs in the terminal, written in Python.

Graph types supported:

- Bar Graphs
- Color charts
- Multi-variable
- Stacked charts
- Histograms
- Horizontal or Vertical
- Emoji!



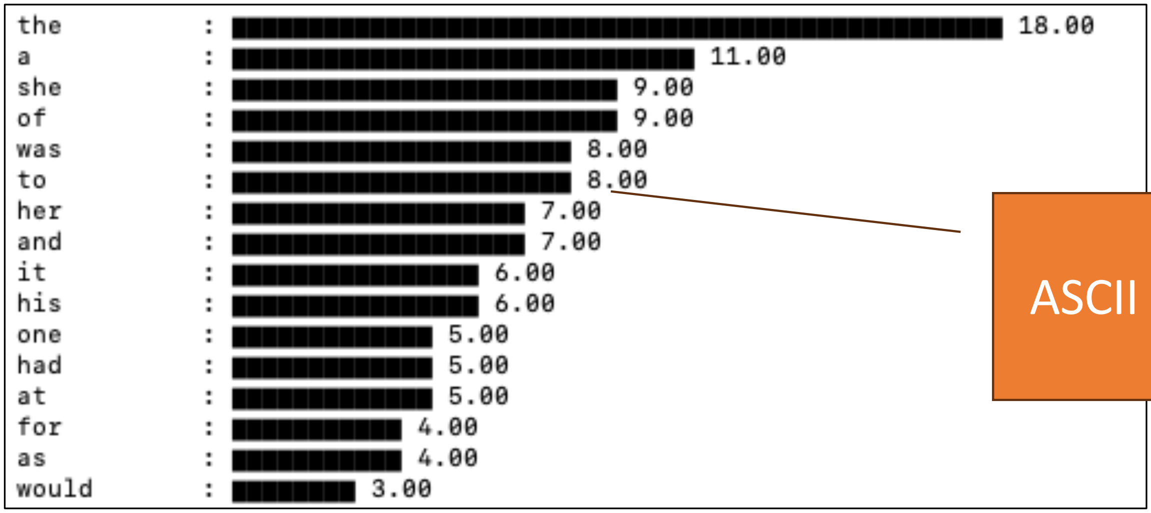
```
termgraph data/ex1.dat

# Reading data from data/ex1.dat

2007: ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 183.32
2008: ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 231.23
2009: ▇ 16.43
2010: ▇▇▇▇ 50.21
2011: ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 508.97
2012: ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 212.05
2014: | 1.00
```

An example using emoji as custom tick:

```
termgraph data/ex1.dat --custom-tick "🏃" --width 20 --title "Running Data"

# Running Data

2007: 🏃🏃🏃🏃🏃🏃🏃 183.32
2008: 🏃🏃🏃🏃🏃🏃🏃🏃🏃 231.23
2009:  16.43
2010: 🏃 50.21
2011: 🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃🏃 508.97
2012: 🏃🏃🏃🏃🏃🏃🏃🏃 212.05
2014:  1.00
```

termgraph

```
cat text.txt | tr '[:punct:]' ' ' | tr ' ' '\n' | sort | uniq
-c | sort -rn | tail -n +2 | awk '{print $2, $1}' | termgraph
```



```
the         : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 18.00
a           : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 11.00
she         : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 9.00
of          : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 9.00
was         : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 8.00
to          : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 8.00
her         : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 7.00
and         : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 7.00
it          : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 6.00
his         : ▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇ 6.00
one         : ▇▇▇▇▇▇▇▇▇▇▇▇▇ 5.00
had         : ▇▇▇▇▇▇▇▇▇▇▇▇▇ 5.00
at          : ▇▇▇▇▇▇▇▇▇▇▇▇▇ 5.00
for         : ▇▇▇▇▇▇▇▇▇▇▇ 4.00
as          : ▇▇▇▇▇▇▇▇▇▇▇ 4.00
would       : ▇▇▇▇▇▇▇▇ 3.00
```

ASCII graphics!

tail –n +2

NAME

tail – display the last part of a file

SYNOPSIS

tail [**–F** | **–f** | **–r**] [**–q**] [**–b** *number* | **–c** *number* | **–n** *number*] [*file ...*]

DESCRIPTION

The **tail** utility displays the contents of *file* or, by default, its standard input, to the standard output.

Numbers having a leading plus (‘+’) sign are relative to the beginning of the input, for example, “–c +2” starts the display at the second byte of the input.

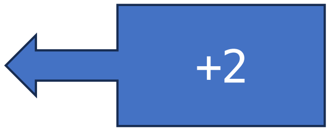
Numbers having a leading minus (‘–’) sign or no explicit sign are relative to the end of the input, for example, “–n 2” displays the last two lines of the input.

–n *number*, **––lines**=*number*

The location is *number* lines.

tail –n +2

```
cat text.txt | tr '[:punct:]' ' ' | tr ' ' '\n' | sort | uniq –c | sort –rn
  86
  18 the
  11 a
   9 she
   9 of
   8 was
   8 to
   7 her
```



+2

awk '{print $2, $1}'

NAME

awk – pattern–directed scanning and processing language

SYNOPSIS

awk [**-F** fs] [**-v** var=value] ['prog' | **-f** progfile] [file ...]

DESCRIPTION

Awk scans each input file for lines that match any of a set of patterns
specified literally in prog or in one or more files specified as **-f** progfile.
With each pattern there can be an associated action that will be performed when
a line of a file matches the pattern.
A pattern–action statement has the form:

pattern **{** action **}**

A missing **{** action **}** means print the line; a missing pattern always matches.
The **print** statement prints its arguments on the standard output
{ print $2, $1 }
Print first two fields in opposite order.

tail –n +2

```
cat text.txt | tr '[:punct:]' ' ' | tr ' ' '\n' | sort | uniq –c | sort
–rn | awk '{print $2, $1}' | termgraph
```

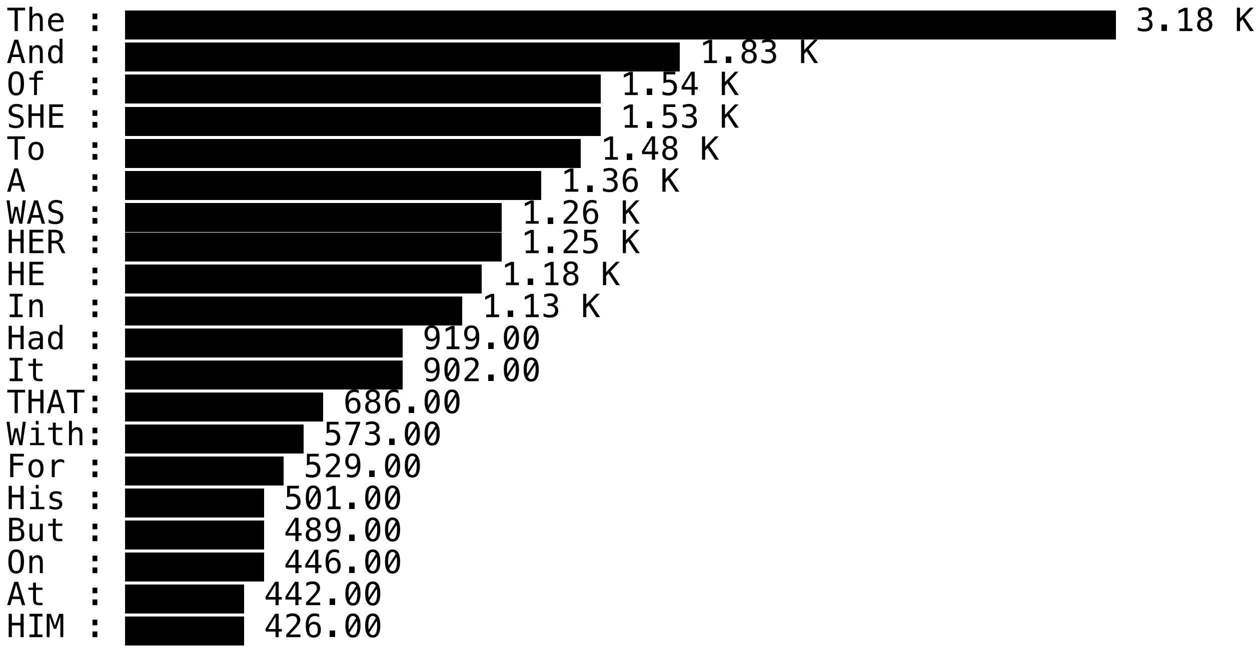
```
86
18 the
11 a
 9 she
 9 of
 8 was
 8 to
 7 her
```

+2

```
the 18
a 11
she 9
of 9
was 8
to 8
her 7
and 7
```

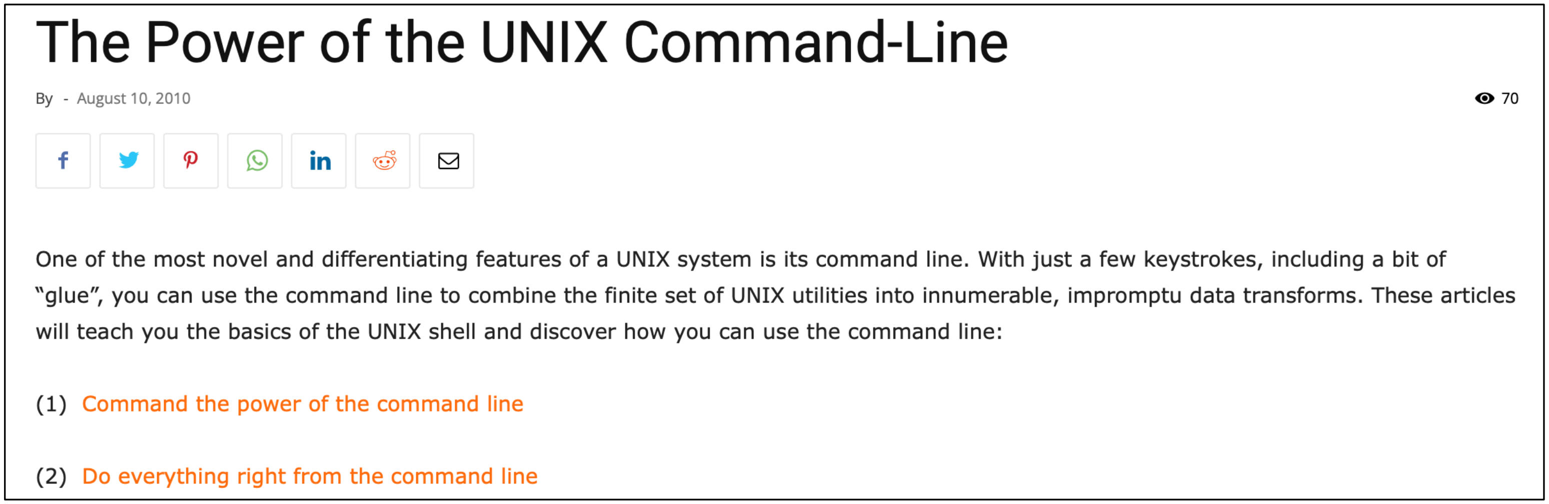
awk '{print $2, $1}'

Mrs. Dalloway



The	3181
And	1835
Of	1540
SHE	1534
To	1476
A	1363
WAS	1261
HER	1254
HE	1179
In	1130
Had	919
It	902
THAT	686
With	573
For	529
His	501
But	489
On	446
At	442
HIM	426

The spirit of Unix (Linux)



The Power of the UNIX Command-Line

By - August 10, 2010

70

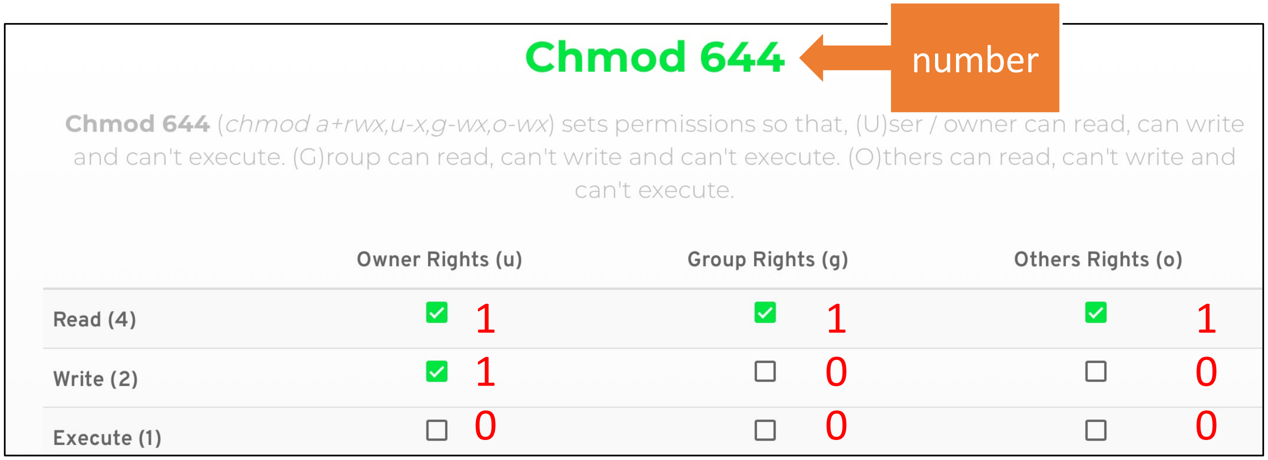
One of the most novel and differentiating features of a UNIX system is its command line. With just a few keystrokes, including a bit of “glue”, you can use the command line to combine the finite set of UNIX utilities into innumerable, impromptu data transforms. These articles will teach you the basics of the UNIX shell and discover how you can use the command line:

(1) Command the power of the command line

(2) Do everything right from the command line

```
wc
sort
uniq
tr
tail
cat
echo
pr
```

Running shell scripts



Chmod 644

number

Chmod 644 (*chmod a+rwx,u-x,g-wx,o-wx*) sets permissions so that, (U)ser / owner can read, can write and can't execute. (G)roup can read, can't write and can't execute. (O)thers can read, can't write and can't execute.

	Owner Rights (u)	Group Rights (g)	Others Rights (o)
Read (4)	☑ 1	☑ 1	☑ 1
Write (2)	☑ 1	□ 0	□ 0
Execute (1)	□ 0	□ 0	□ 0

Command:

- `chmod` *`permissions filename`*
- *`permissions`*`:` e.g. u+x (*user add execute*) or a number

Recall everything is binary:

- 110 = 6, 100 = 4
- 644 = 110100100 (*3 groups of binary*)