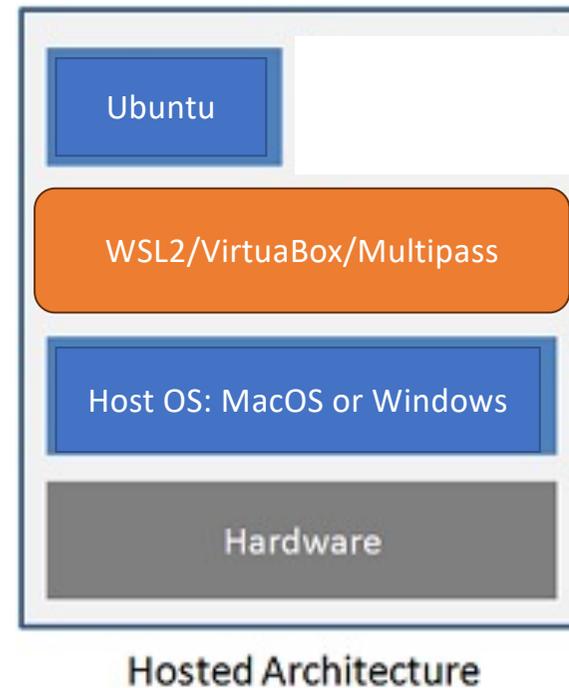


Lecture 5

408/508 *Computational  
Techniques for Linguists*

## Last Time

- Installing Ubuntu 22.04 LTS inside WSL2/VirtualBox/Multipass.
- *did everyone succeed?*
- If you didn't,
  - you can use **Terminal** directly (*but it's not sandboxed*)



# Today's Topics

- Terminal commands in Ubuntu

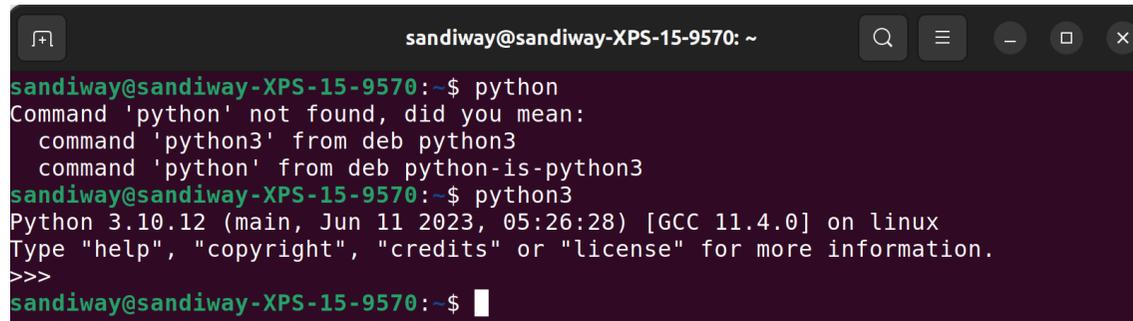
# Ubuntu: bash

bash contains a programming language!

- **Terminal:**

- runs a **shell**: typically **bash** (or a lightweight clone of bash, e.g. **zsh** on macOS)
- enter commands:
  - some are built-in to the shell (programming language),
  - others are executable files in specified directories (\$PATH),
  - still others will require package `apt-get install` or `apt install`
  - Example:

Type Control-D  
to exit



```
sandiway@sandiway-XPS-15-9570: ~  
sandiway@sandiway-XPS-15-9570:~$ python  
Command 'python' not found, did you mean:  
  command 'python3' from deb python3  
  command 'python' from deb python-is-python3  
sandiway@sandiway-XPS-15-9570:~$ python3  
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
sandiway@sandiway-XPS-15-9570:~$
```

# Ubuntu: bash

Decimal	Hex	Char
0	0	[NULL]
1	1	[START OF HEADING]
2	2	[START OF TEXT]
3	3	[END OF TEXT]
4	4	[END OF TRANSMISSION]
5	5	[ENQUIRY]
6	6	[ACKNOWLEDGE]
7	7	[BELL]
8	8	[BACKSPACE]
9	9	[HORIZONTAL TAB]
10	A	[LINE FEED]
11	B	[VERTICAL TAB]
12	C	[FORM FEED]
13	D	[CARRIAGE RETURN]
14	E	[SHIFT OUT]
15	F	[SHIFT IN]
16	10	[DATA LINK ESCAPE]
17	11	[DEVICE CONTROL 1]
18	12	[DEVICE CONTROL 2]
19	13	[DEVICE CONTROL 3]
20	14	[DEVICE CONTROL 4]
21	15	[NEGATIVE ACKNOWLEDGE]
22	16	[SYNCHRONOUS IDLE]
23	17	[ENG OF TRANS. BLOCK]
24	18	[CANCEL]
25	19	[END OF MEDIUM]
26	1A	[SUBSTITUTE]
27	1B	[ESCAPE]
28	1C	[FILE SEPARATOR]
29	1D	[GROUP SEPARATOR]
30	1E	[RECORD SEPARATOR]
31	1F	[UNIT SEPARATOR]

- What is Control-D?
  - *why does it quit?*

Non-Printing Characters				
Name	Ctrl char	Dec	Hex	Char
null	ctrl-@	0	00	NUL
start of heading	ctrl-A	1	01	SOH
start of text	ctrl-B	2	02	STX
end of text	ctrl-C	3	03	ETX
end of xmit	ctrl-D	4	04	EOT
enquiry	ctrl-E	5	05	ENQ
acknowledge	ctrl-F	6	06	ACK
bell	ctrl-G	7	07	BEL

Well, let's ring the bell!

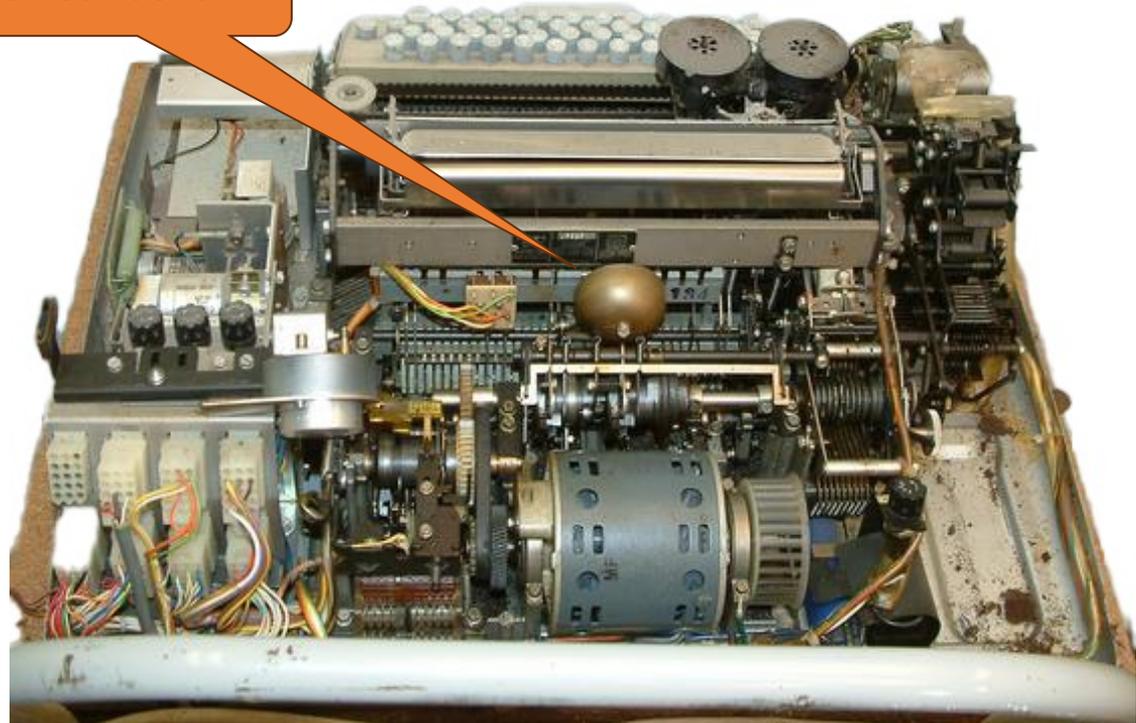
# Ubuntu: bash

**Teletype  
ASR 33**

1974 price:  
\$755 to \$1220  
Today:  
\$7550



a real bell!



[pdp8online.com](http://pdp8online.com)

# Ubuntu: bash



- BTW:
  - 1974 price: \$1220
  - Today: \$7550



### Customize your Mac Studio

Apple M2 Ultra with 24-core CPU, 76-core GPU, 32-core Neural Engine  
192GB unified memory  
4TB SSD storage

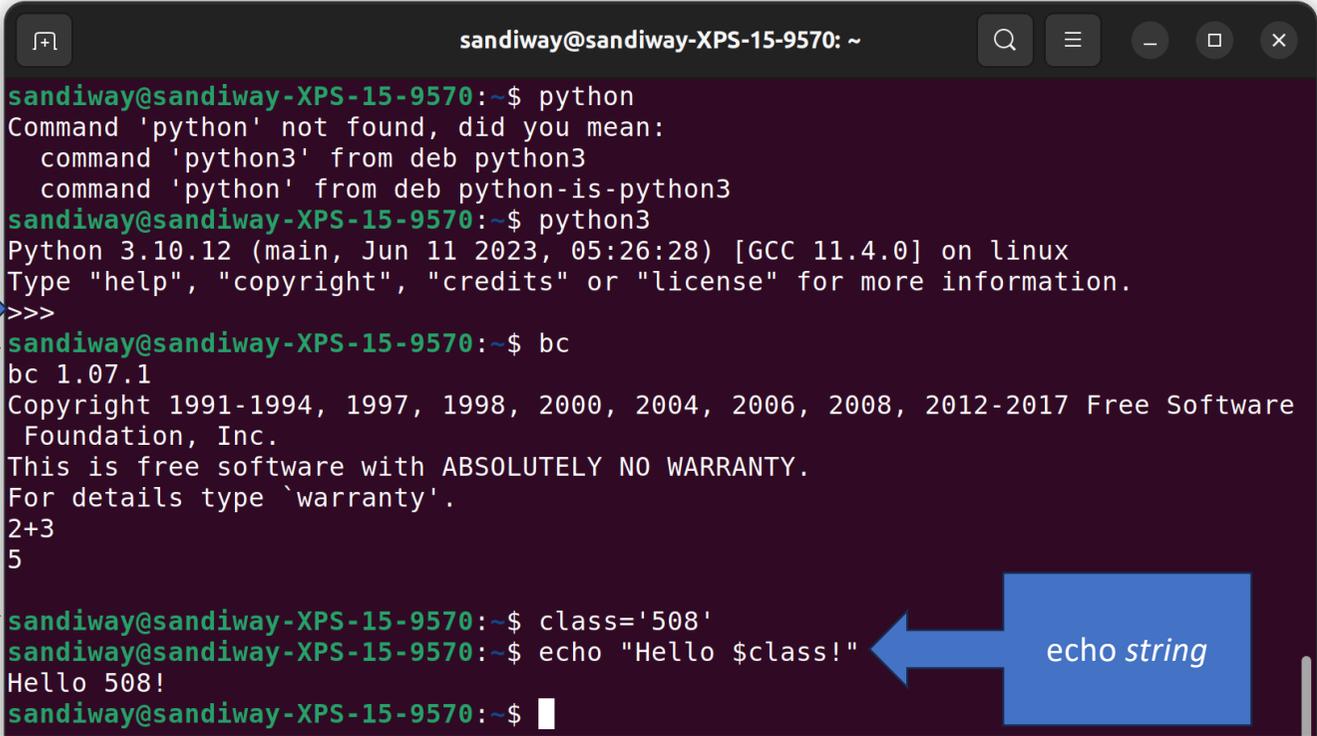
Front: Two Thunderbolt 4 ports, one SDXC card slot  
Back: Four Thunderbolt 4 ports, two USB-A ports, one HDMI port, one 10Gb Ethernet port, one 3.5 mm headphone jack

 Order today. Delivers to **85701**  
Sep 13 - Sep 15 — Free

 Order now. Pick up, in store:

**\$7,599.00 or** [Continue](#)   
**\$633.25/mo. for 12 mo.\***

# Ubuntu: bash



```
sandiway@sandiway-XPS-15-9570: ~  
sandiway@sandiway-XPS-15-9570:~$ python  
Command 'python' not found, did you mean:  
  command 'python3' from deb python3  
  command 'python' from deb python-is-python3  
sandiway@sandiway-XPS-15-9570:~$ python3  
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
sandiway@sandiway-XPS-15-9570:~$ bc  
bc 1.07.1  
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software  
Foundation, Inc.  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type `warranty'.  
2+3  
5  
sandiway@sandiway-XPS-15-9570:~$ class='508'  
sandiway@sandiway-XPS-15-9570:~$ echo "Hello $class!"  
Hello 508!  
sandiway@sandiway-XPS-15-9570:~$
```

Control-D to exit

bc calculator

class shell variable

echo string

# Ubuntu: bash

- **Terminal:**

- there's a (*customizable*) prompt
- command history is editable (up-arrow to retrieve previous command ...)
- pre-defined environment variables: **env**
- lots of packages are pre-loaded: **wish**, **python**, **perl**, etc.
- **dpkg** (package manager for Debian)
- **man** *command-name* (brings up manual page)

`dpkg -l (list)`

# Ubuntu: bash

- simple commands:

- **pwd**

print working directory

- **ls (ls -a)**

list current directory

(-a option: show . (dot) files too)

- **cd**

change directory

- **mkdir**

create a new directory

- **which** *name*

prints the directory where command

*name* is located, or nothing if it can't be

found in the PATH

- **man** *name*

display manual page for command *name*

- **echo \$SHELL**

prints the shell (\$ prefixes a variable)

- **echo \$PATH**

shows the directories where the shell will

look for commands

# Ubuntu: bash

## Directory shortcuts (abbreviations):

- your home directory: `~`
- current directory: `.`
- parent directory: `..`
- Examples:
  - `cd ..` (go to parent directory)
  - `mkdir ~/tmp` (create a new directory called tmp in your home directory)
  - `touch tmp` (create a new file tmp in the current directory if tmp doesn't already exist, or update the timestamp)
  - `ls -l tmp` (list attributes of file tmp in long format)

```
-rw-r--r-- 1 sandiway staff 0 Sep 4 09:26 tmp
```

permissions  
user group all  
r = read  
w = write  
x = execute

owner

group size  
(bytes)

date  
time  
modified

filename

# Shell Arithmetic (very primitive -> less rigid)

- at the shell prompt:

1. `expr 1 + 3`

2. `expr 2 '*' 2`

3. `echo `expr 7 + 3``

(Need spaces cf. `expr 1+3`)

(cf. `expr 2 * 2`)



```
sandiway@sandiway-XPS-15-9570: ~  
sandiway@sandiway-XPS-15-9570:~$ expr 1+3  
1+3  
sandiway@sandiway-XPS-15-9570:~$ expr 1 + 3  
4  
sandiway@sandiway-XPS-15-9570:~$ expr 2 '*' 2  
4  
sandiway@sandiway-XPS-15-9570:~$ expr 2 * 2  
expr: syntax error: unexpected argument '0200991.txt'  
sandiway@sandiway-XPS-15-9570:~$
```

\* is a file wildcard character

# Shell Arithmetic (very primitive -> less rigid)

- at the shell prompt:

4. `i=2`

(NO SPACES! cf. `i = 2`)

5. `expr $i + 1`

6. `let x=1+3`

(cf. `let x=1 + 3`)

7. `echo $x`

8. `let i=$i+1`

(also ok `let i=i+1`)

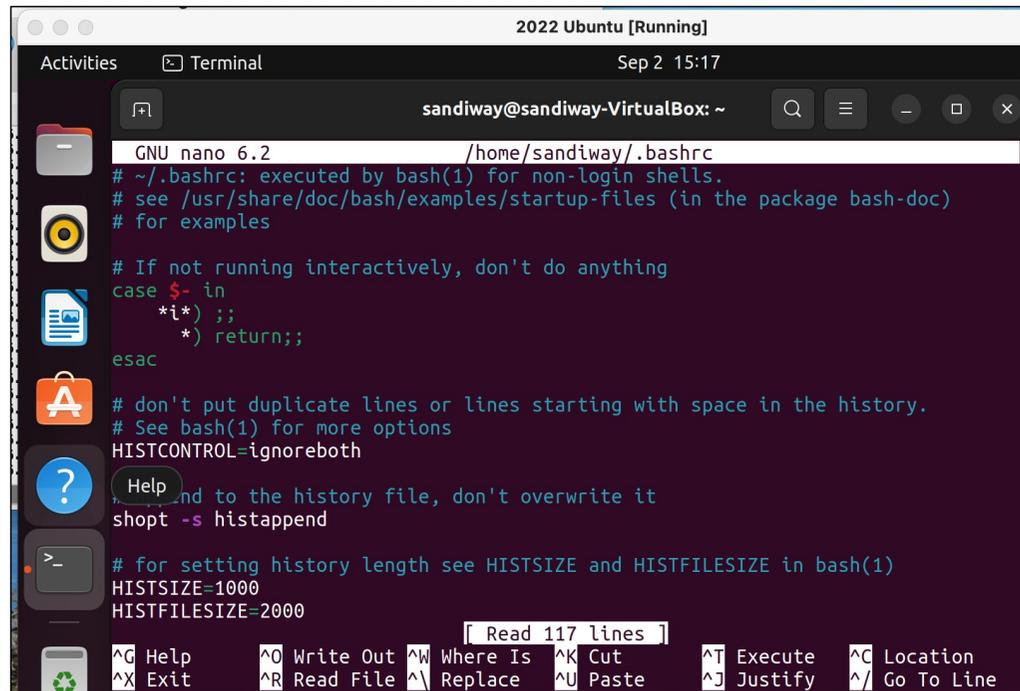
9. `echo $i`

# Shell Arithmetic (very primitive -> less rigid)

- at the shell prompt `((...))` – easier to use, less rigid syntax:
  10. `i=$((1+3))` (set variable `i` to value of expr `1 + 3`)
  11. `((x = 1+ 3))` (spaces not significant)
  12. `echo $x`
  13. `echo $((1+3))`
  14. `((i=i+1))` (also ok `let i=$((i+1))`)

# Ubuntu

nano ~/.bashrc (run the nano text editor on the startup file)



The screenshot shows a terminal window titled "2022 Ubuntu [Running]" with the date and time "Sep 2 15:17". The terminal is running the nano text editor on the file "/home/sandiwai/.bashrc". The editor's title bar shows "GNU nano 6.2 /home/sandiwai/.bashrc". The content of the file is as follows:

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *(i) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

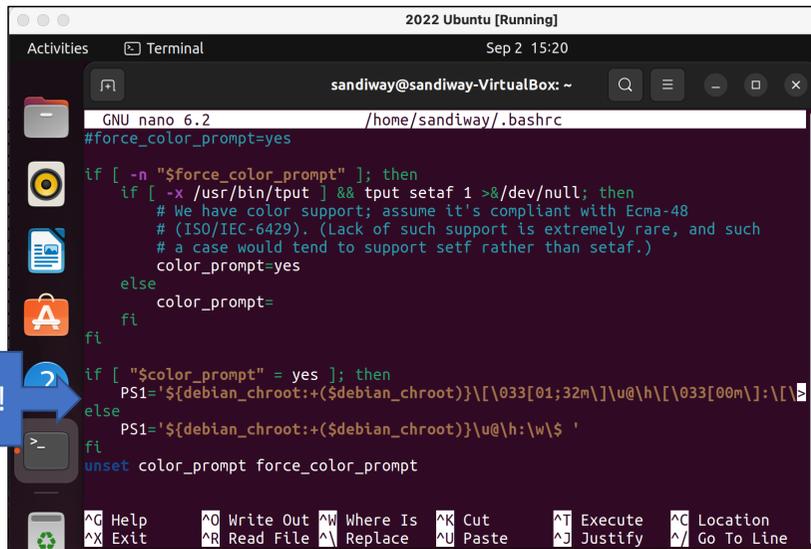
# Help and to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
```

At the bottom of the terminal, there is a status bar with the text "Read 117 lines" and a list of keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^\_ Go To Line.

# Ubuntu

scroll down (arrow keys) until you see PS1=



```
GNU nano 6.2 /home/sandiwai/.bashrc
#force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
  if [ -x /usr/bin/tput ] && tput setaf 1 >/dev/null; then
    # We have color support; assume it's compliant with Ecma-48
    # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
    # a case would tend to support setf rather than setaf.)
    color_prompt=yes
  else
    color_prompt=
  fi
fi

if [ "$color_prompt" = yes ]; then
  PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;32m\]\w\$ '
else
  PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt

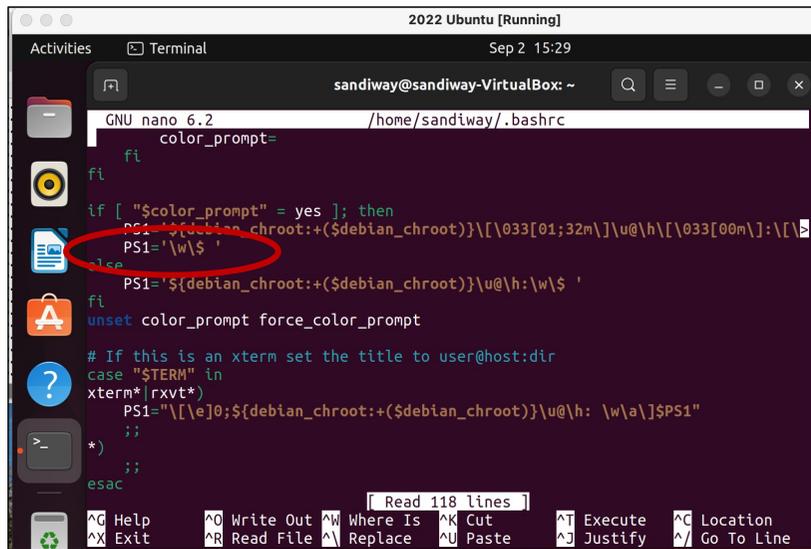
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^_ Replace    ^U Paste     ^J Justify   ^_ Go To Line
```

- let's insert a line `PS1= '\w\$ '`
  - PS1 is the variable holding the prompt string
  - '\w\\$ ' is a single-quoted string
  - \w means print working directory
  - \\$ means print \$
- Then Control-O RETURN (to write out the change)
- Then Control-X to Exit Nano
- Then close the Terminal and restart it

# Ubuntu

modified `~/ .bashrc` file (~ means your home directory)

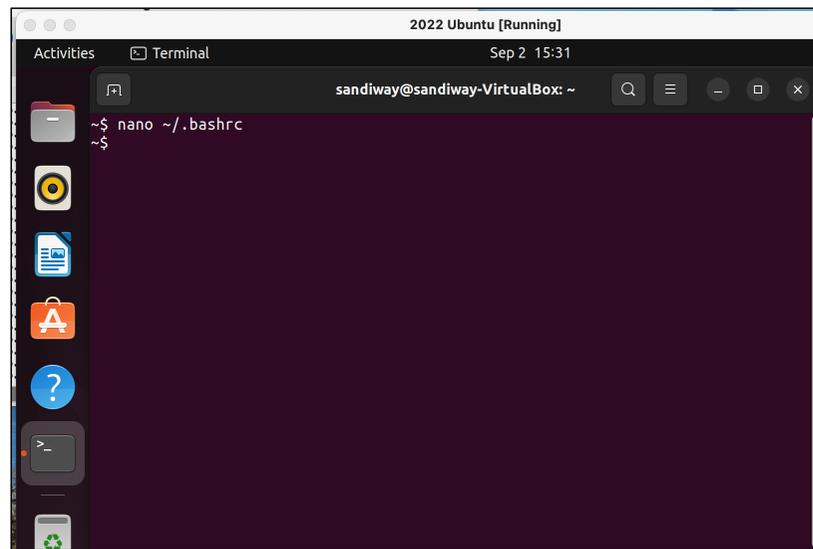
`~$` is a nice and short prompt!



```
GNU nano 6.2 /home/sandiway/.bashrc
color_prompt=
fi
if [ "$color_prompt" = yes ]; then
PS1='\w\$ '
else
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
PS1="[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
;;
*)
;;
esac

Read 118 lines
Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line
```



```
~$ nano ~/.bashrc
~$
```

# Shell program

{1..10..2} means range from 1 to 10 incrementing by 2  
; (semicolon) OR newline terminates/separates statements

```
2022 Ubuntu [Running]
Activities Terminal Sep 2 15:41
sandiwai@sandiwai-VirtualBox: ~
~$ nano ~/.bashrc
~$ echo $SHELL
/bin/bash
~$ for i in {1..10}
> do
> echo "$i time through the loop"
> done
1 time through the loop
2 time through the loop
3 time through the loop
4 time through the loop
5 time through the loop
6 time through the loop
7 time through the loop
8 time through the loop
9 time through the loop
10 time through the loop
~$
```

← echo means print

```
2022 Ubuntu
Activities Terminal S
sandiwai@sandiwai-Vi
~$ for i in {1..10..2}; do echo "$i"; done
1
3
5
7
9
~$
```

```
~$ for ((i=1; i<=10; i=i+2)); do echo "$i by 2"; done
1 by 2
3 by 2
5 by 2
7 by 2
9 by 2
```

macOS