

Command-line interface (CLI)

Prof. Pai H. Chou
National Tsing Hua University

Outline

- What is CLI
- CLI for Unix-like shells
- The prompt
- Files and directories
- Summary

What is Command-Line Interface?

- Text-based user interface
 - as opposed to graphical user interface (GUI)
 - type text commands rather than point-and-click with a mouse
- Purposes
 - viewing and manipulating files and directories
 - running and controlling application programs

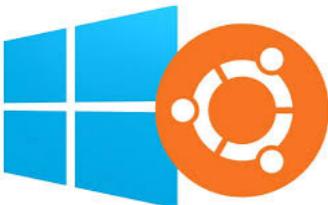
Most systems have a CLI

- Windows
 - Built-in: DOS shell.
 - but we install and use a Unix-like shell (Bash) to be consistent with other systems
 - Cygwin (all versions - recommended), or
 - WSL (Windows 10 only)
- Linux, macOS (MacOS X)
 - most likely already installed
 - search for the Terminal application program

Options for Windows

- All versions
 - Cygwin (www.cygwin.com) - a Linux-like environment inside Windows
 - VMware Player (free) - virtual machine
<https://my.vmware.com/web/vmware/downloads>
- Windows 10
 - Bash on Ubuntu on Windows 10 (also known as Windows Subsystem for Linux - WSL):
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Options for Windows

Environment	What is it	trade-offs
	Cygwin Linux-like environment	least resource; install packages as needed; runs on older Windows
	Vmware, Virtual Box virtual machine	needs most resource, complete Linux, free for education (Vmware player)
	WSL Linux subsystem	Windows 10 only; needs to install a few packages

Install Cygwin (1/4)

1. Download installer

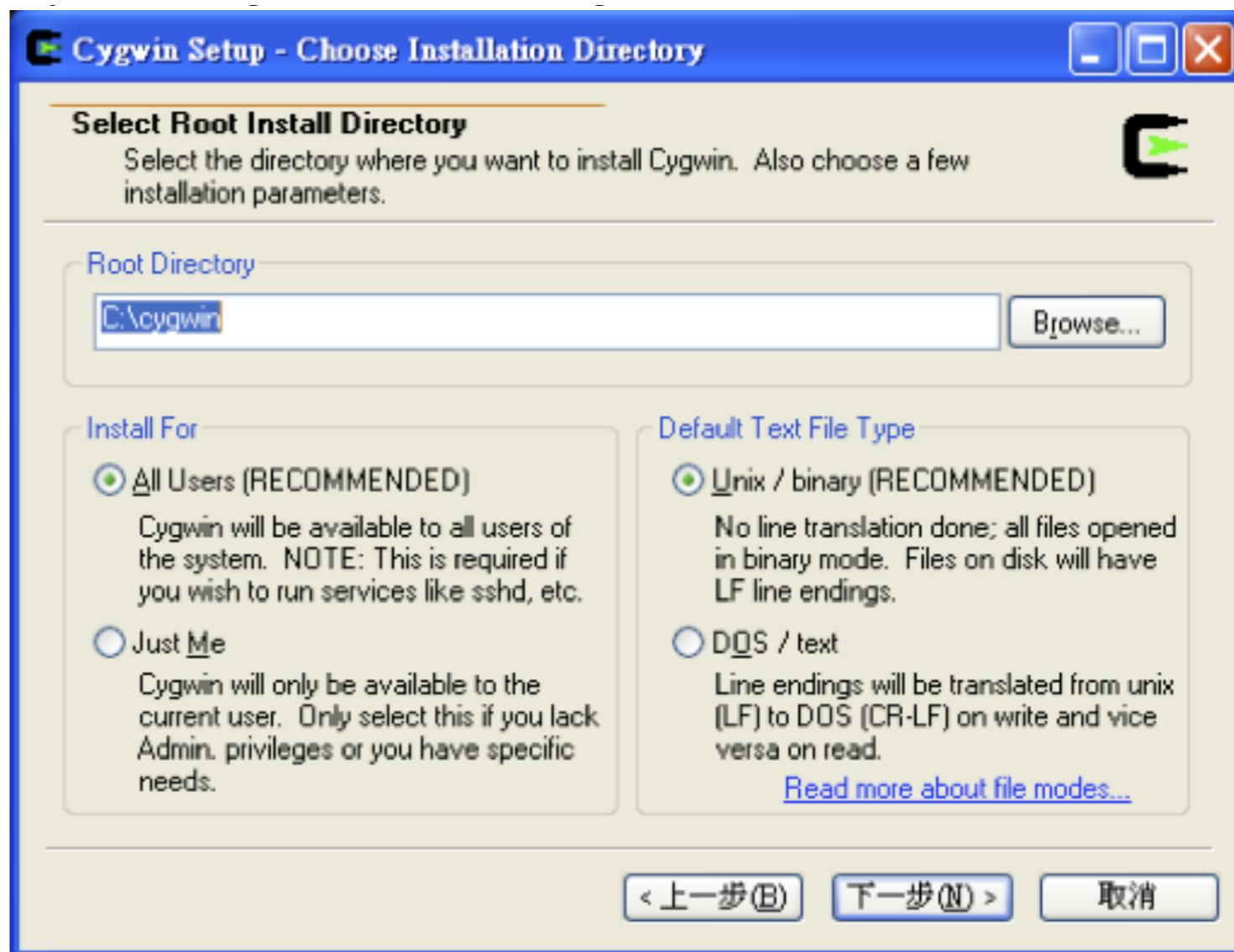
www.cygwin.com



2. Run setup.exe

1. choose installation directory

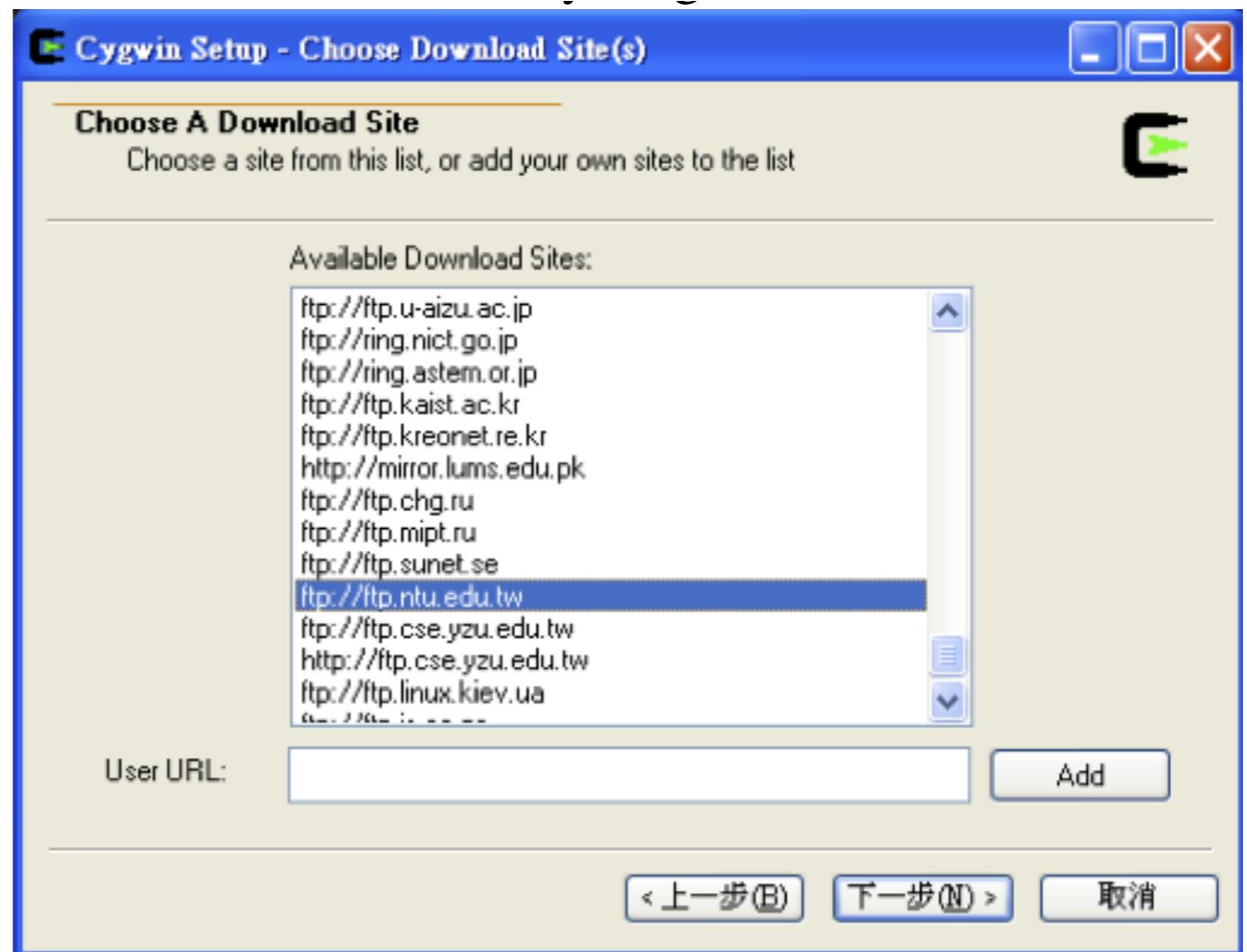
2. click Next



Install Cygwin (2/4)

3. Choose a download site

- a domestic one may be faster
- click Next



Install Cygwin (3/4)

3. Search and Select packages to install

1. Python3

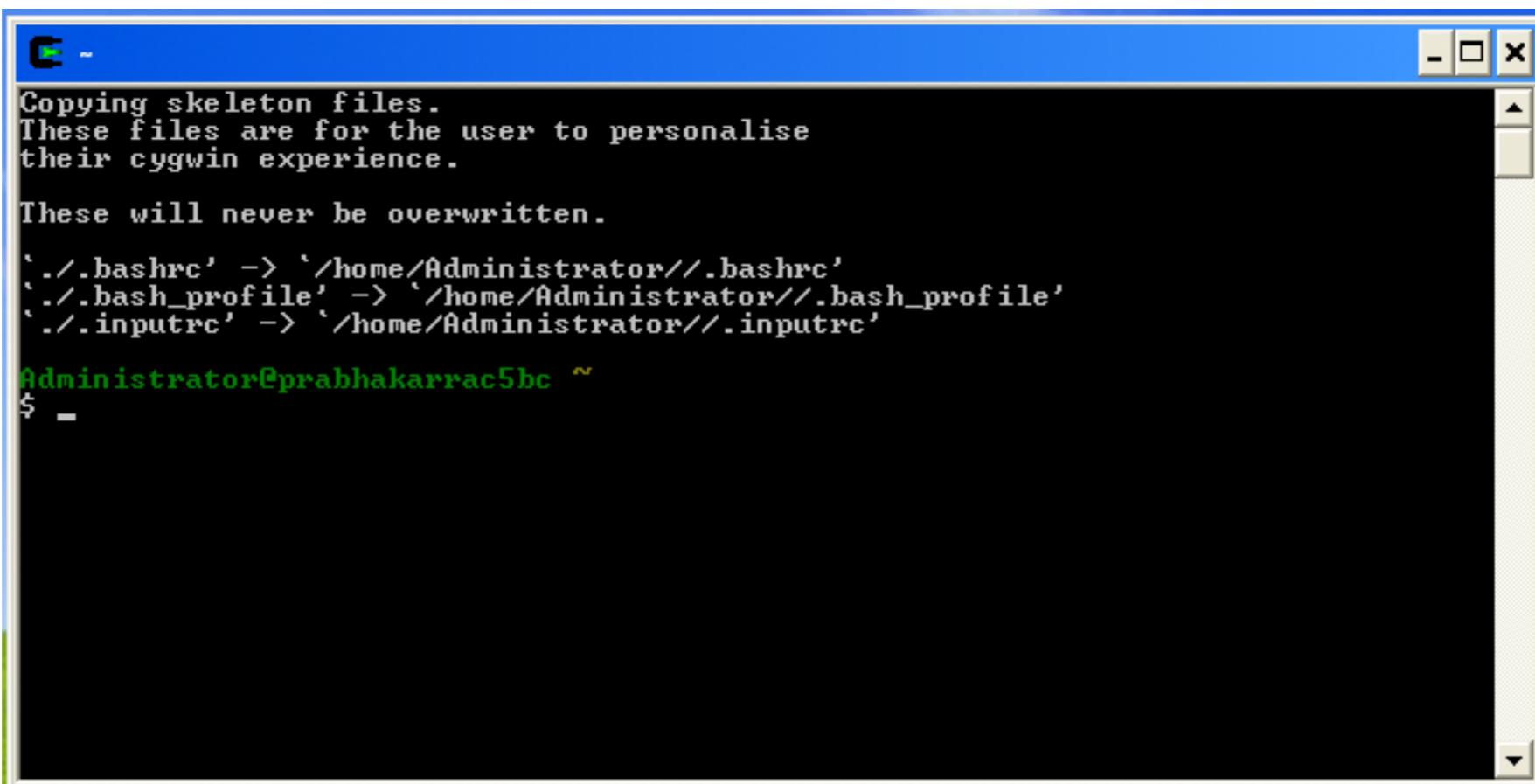
2. vim

- you can always come back here and install or uninstall packages
- click Next to install.
This may take a while!



Install Cygwin (4/4)

- After installation is complete, you get a shortcut on your desktop named Cygwin.
 - when open, it runs bash. It is your command-line interface



The screenshot shows a Cygwin terminal window titled 'C -'. The window displays the following text:

```
Copying skeleton files.
These files are for the user to personalise
their cygwin experience.

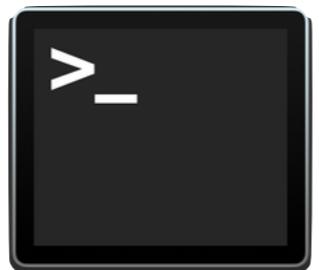
These will never be overwritten.

'./.bashrc' -> '/home/Administrator/.bashrc'
'./.bash_profile' -> '/home/Administrator/.bash_profile'
'./.inputrc' -> '/home/Administrator/.inputrc'

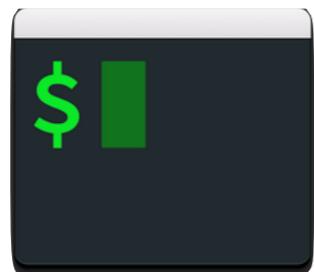
Administrator@prabhakarrac5bc ~
$ -
```

macOS (1/3): terminal

- already preinstalled with a terminal
 - Applications > Utilities > Terminal.app

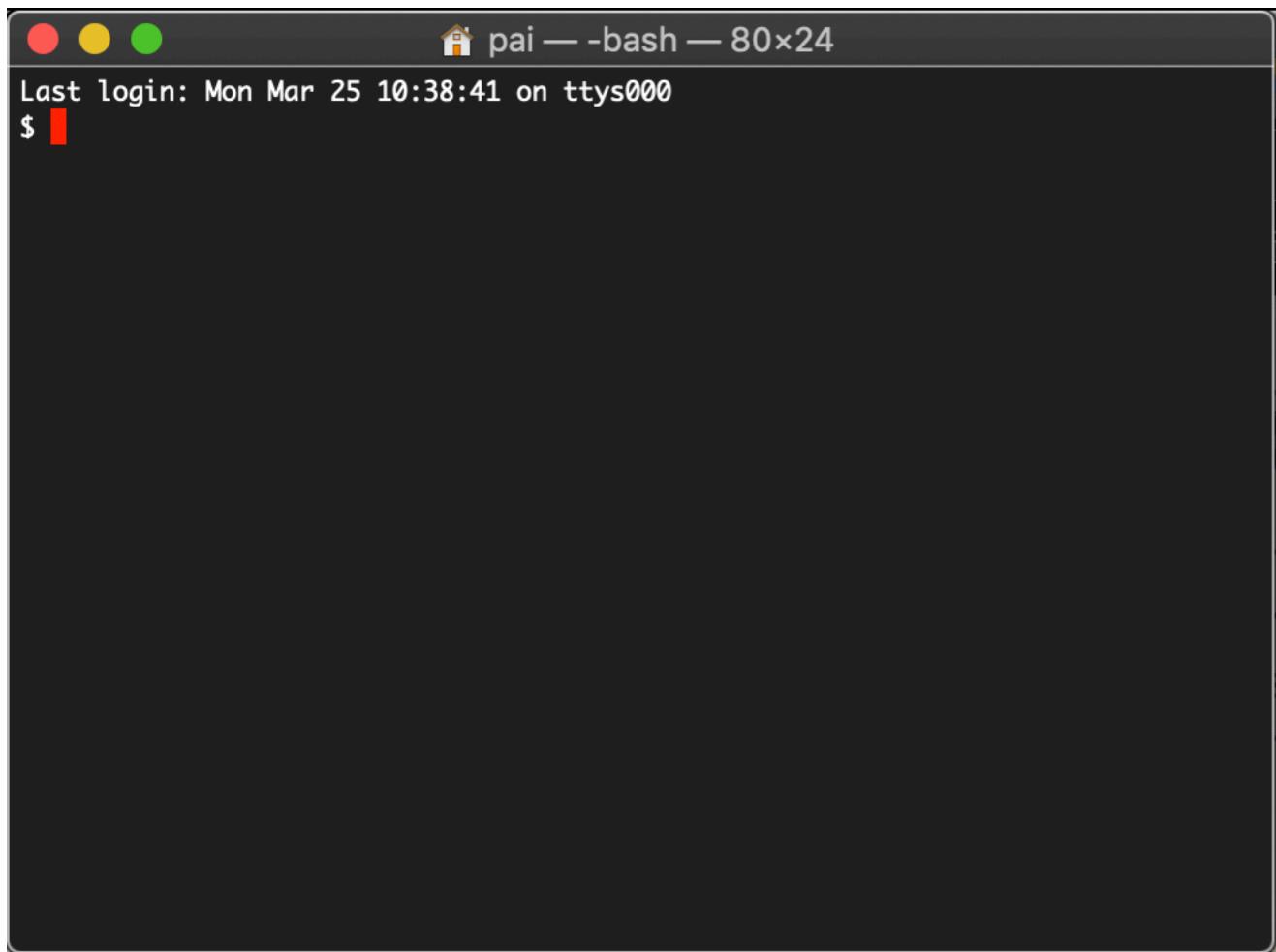


- Alternative: iTerm.app
 - Download from www.iterm2.com



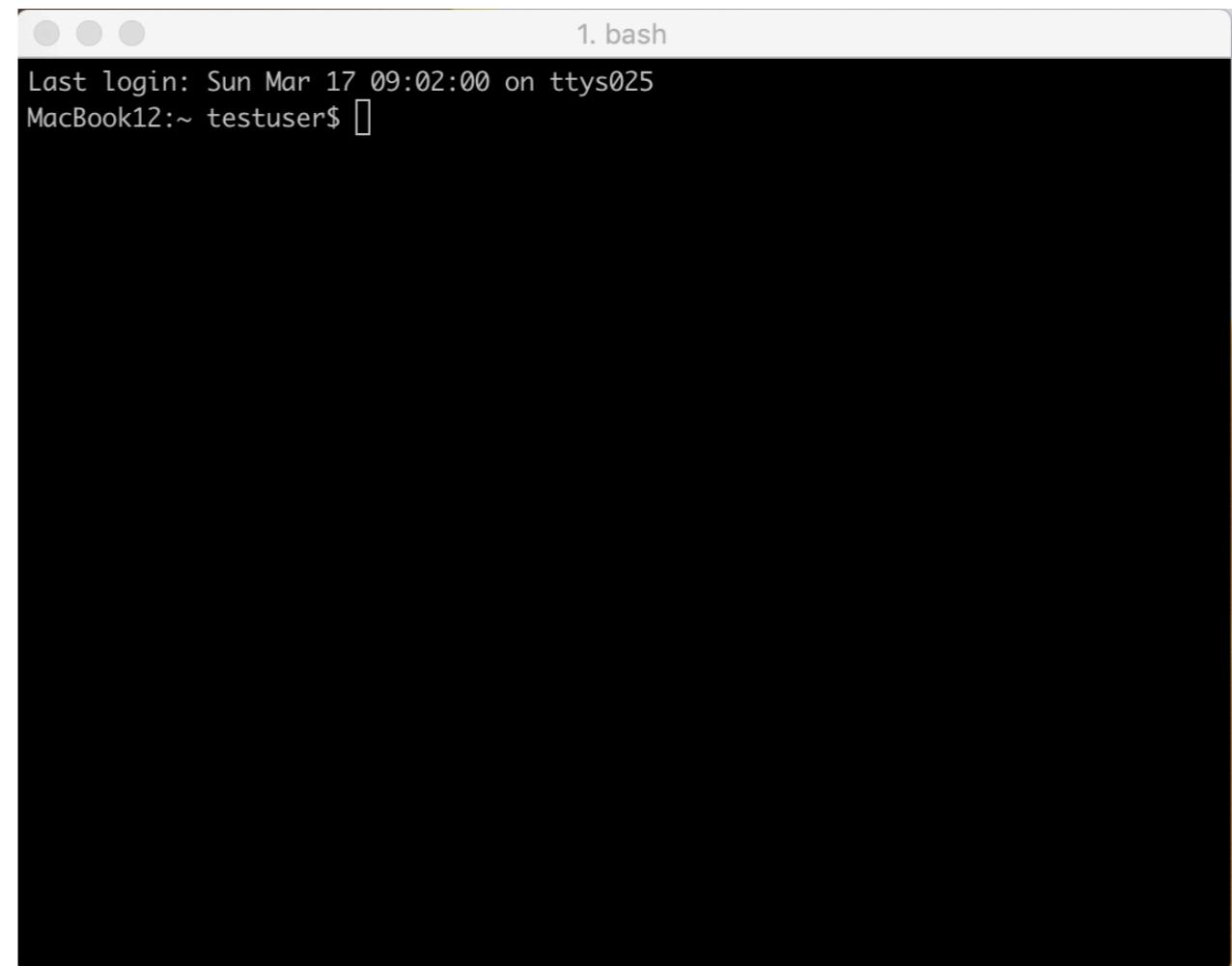
macOS (2/3) Terminal or iTerm

- Terminal.app



A screenshot of the macOS Terminal.app window. The title bar reads "pai — -bash — 80x24". Below it, the status bar shows "Last login: Mon Mar 25 10:38:41 on ttys000". The main terminal area has a black background and a red cursor at the bottom left. The prompt "\$" is visible at the bottom.

- iTerm



A screenshot of the iTerm window. The title bar reads "1. bash". Below it, the status bar shows "Last login: Sun Mar 17 09:02:00 on ttys025" and "MacBook12:~ testuser\$". The main terminal area has a light gray background and a white cursor at the bottom right. The prompt "\$" is visible at the bottom.

- either one works fine. iTerm may have more features.
- type `vi` or `vim` to run vim text editor, similar to Linux

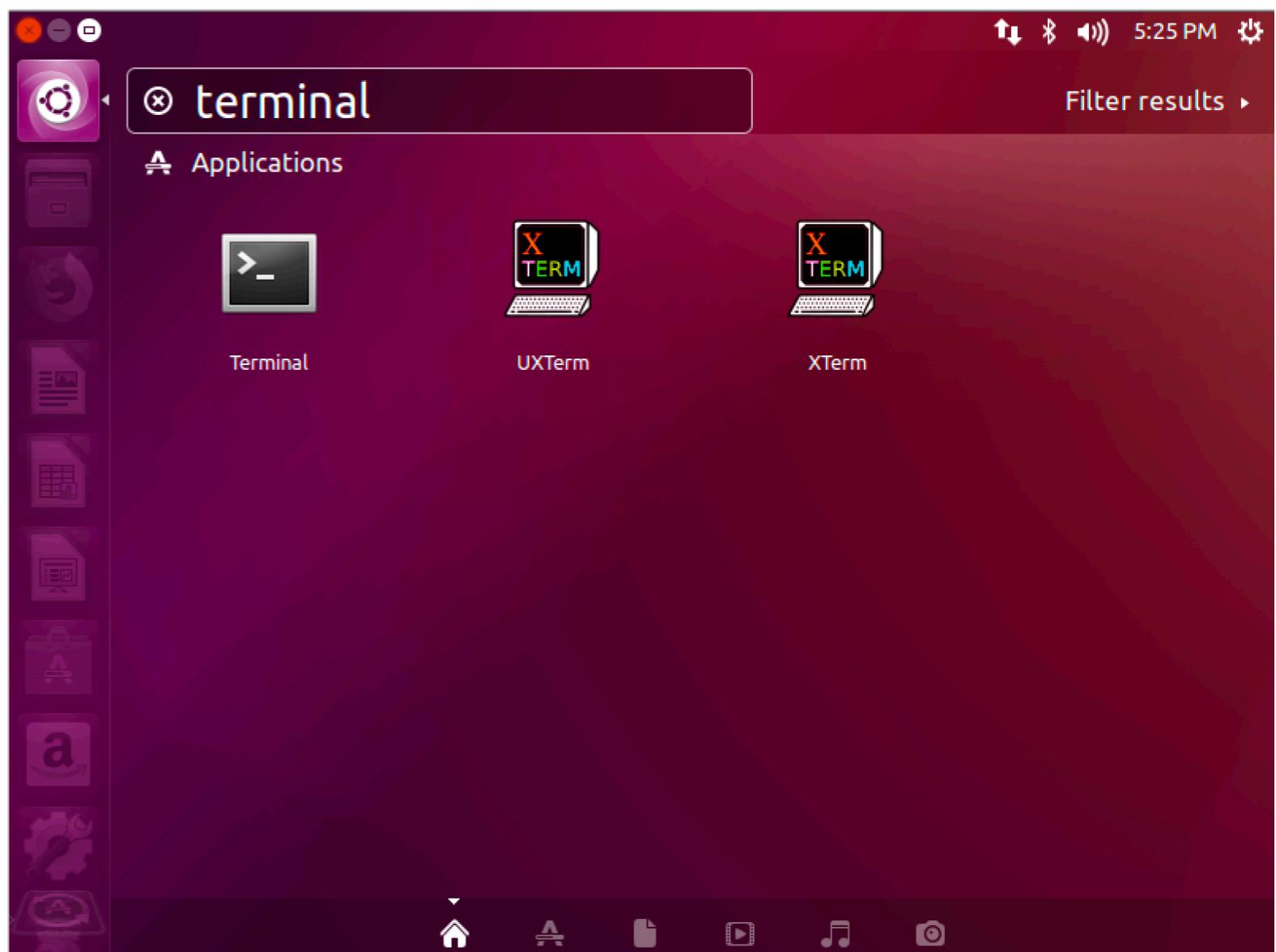
macOS (3/3): Command-line Tools

- Python3 and related command-line tools should have been installed
- Optional, install command line tools
 - type `xcode-select --install` into bash terminal
- Recommend: install home-brew package manager
 - <https://brew.sh>
 - `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`



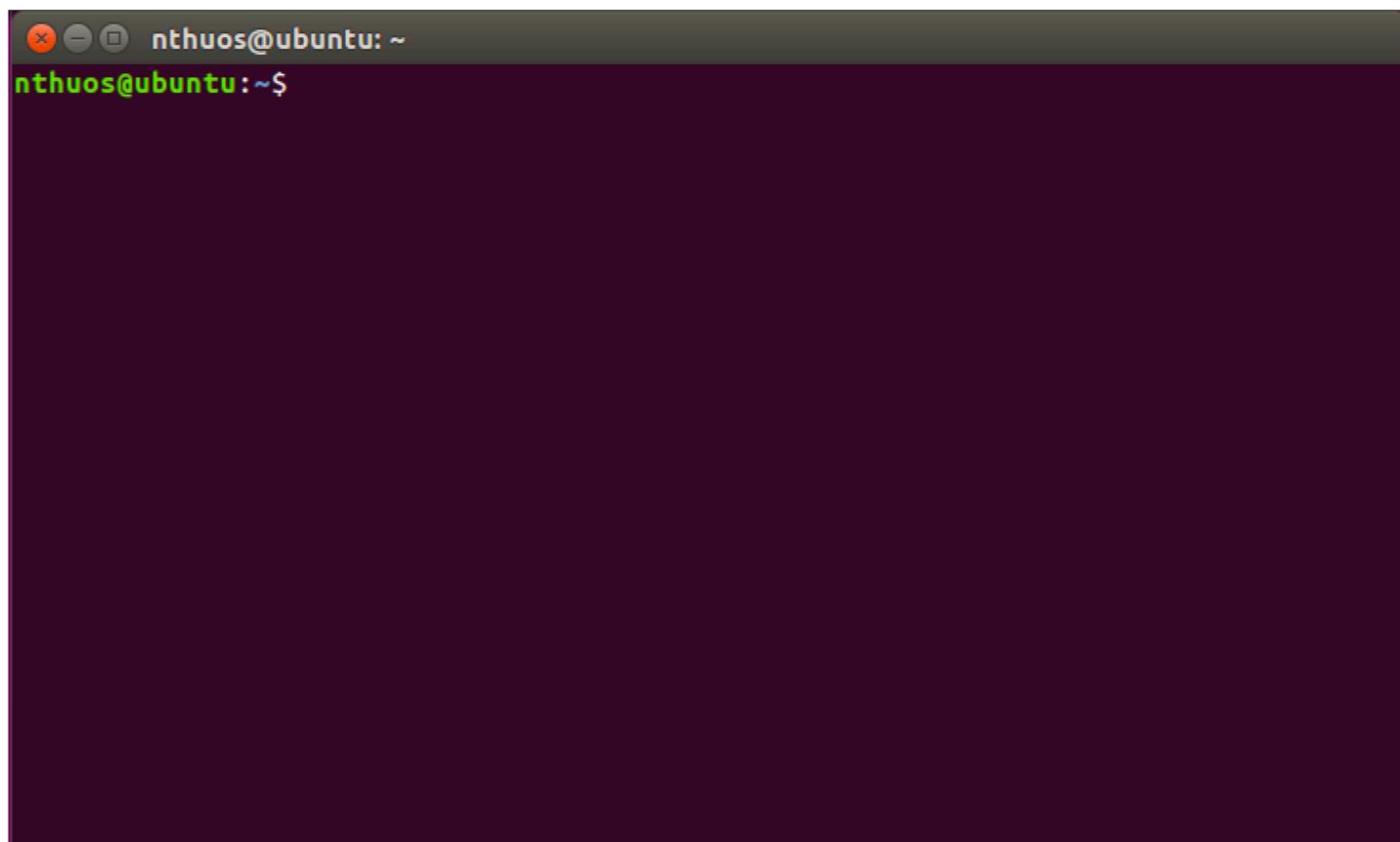
Linux (1/5): terminal

- Example:
Ubuntu 16.04
 - search for Terminal
 - double-click to launch it



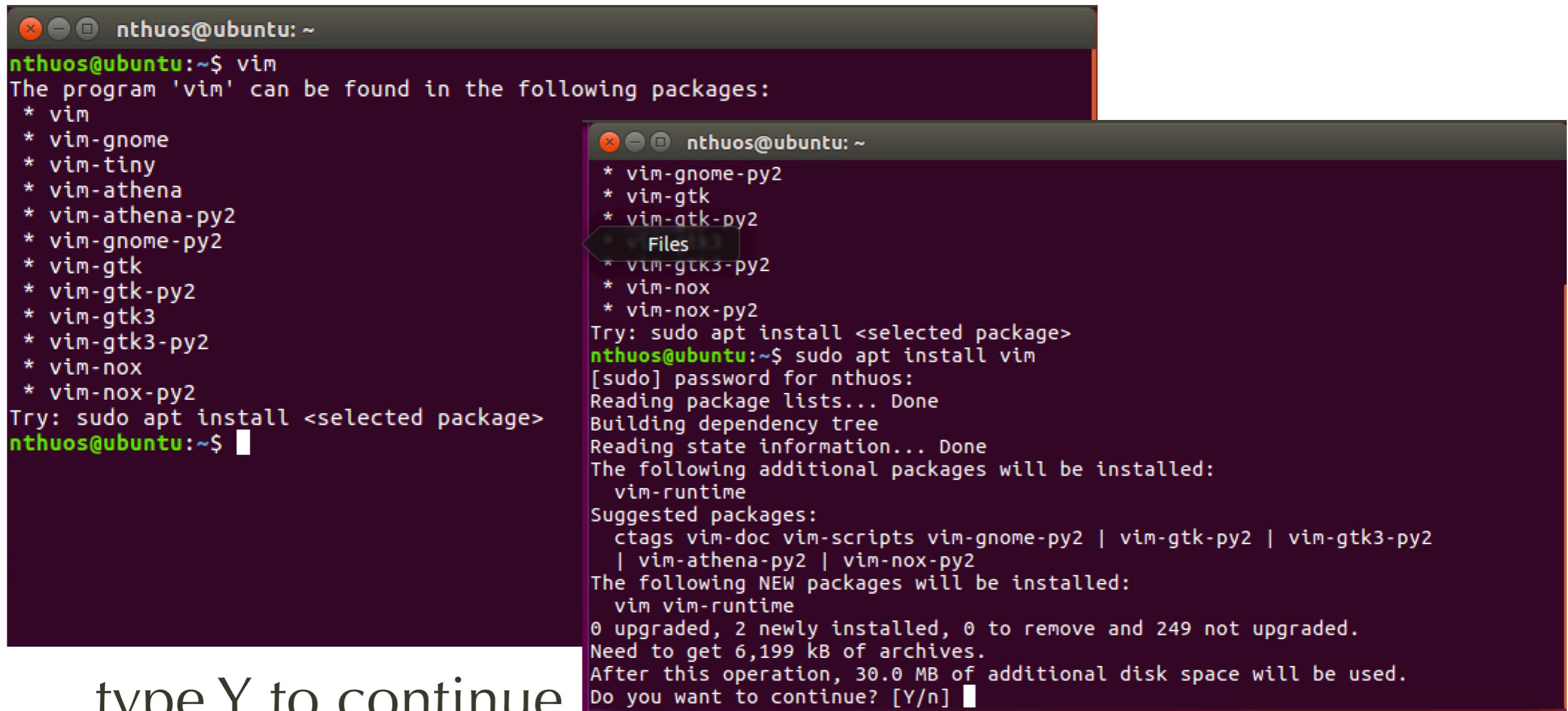
Linux (2/5): bash in terminal

- Terminal runs bash by default
 - prompt may look like *user@ubuntu:~\$*
 - in this example, user is nthuos, so the prompt looks like *nthuos@ubuntu:~\$*
 - ~ means home directory.



Linux (3/5): install vim

- type vim
 - not installed: type sudo apt install vim



The terminal window shows the following output:

```
nthuos@ubuntu:~$ vim
The program 'vim' can be found in the following packages:
 * vim
 * vim-gnome
 * vim-tiny
 * vim-athena
 * vim-athena-py2
 * vim-gnome-py2
 * vim-gtk
 * vim-gtk-py2
 * vim-gtk3
 * vim-gtk3-py2
 * vim-nox
 * vim-nox-py2
Try: sudo apt install <selected package>
nthuos@ubuntu:~$
```

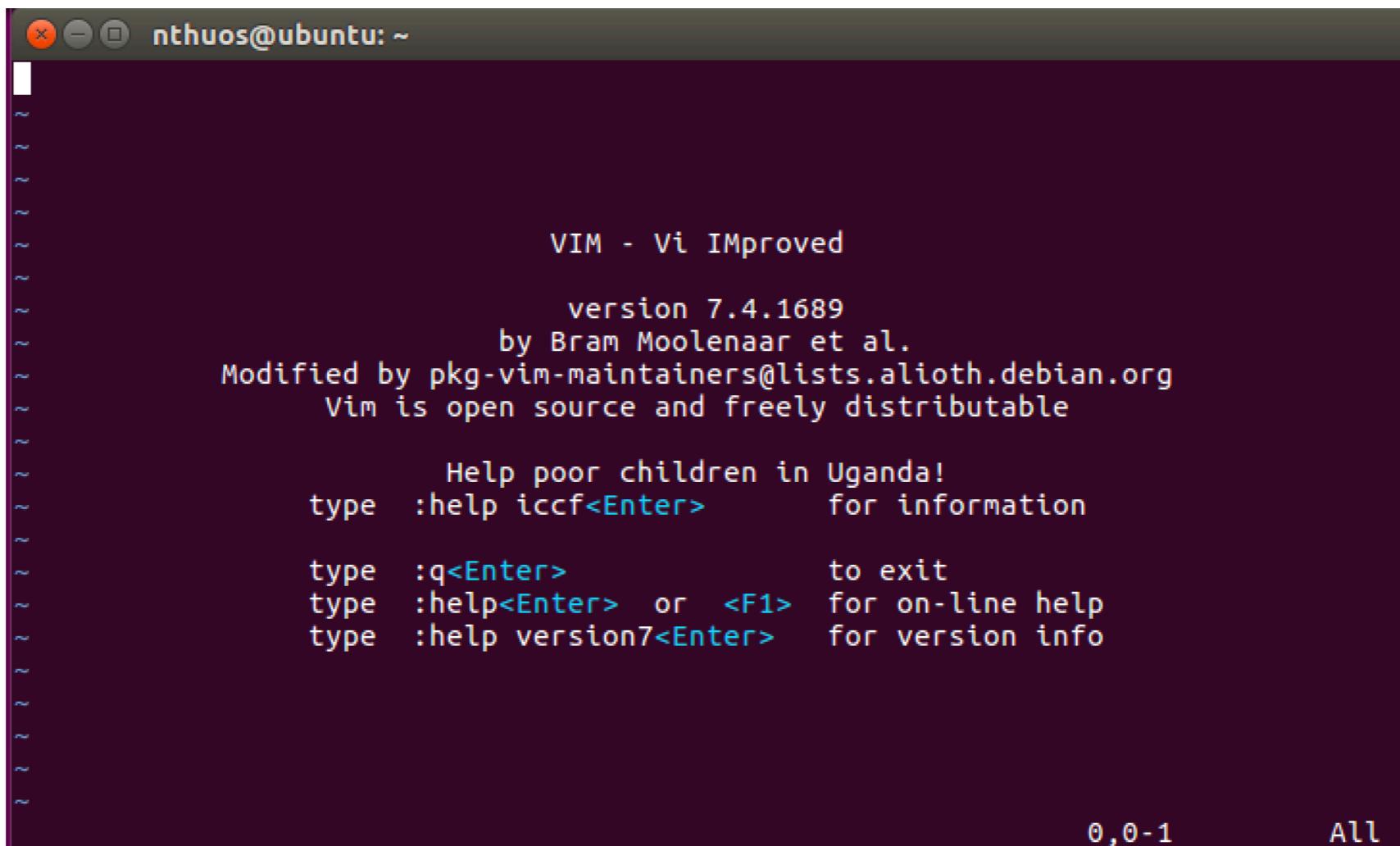
The terminal window then shows the results of the apt search command for 'vim':

```
* vim-gnome-py2
* vim-gtk
* vim-atk-py2
Files
* vim-gtk3-py2
* vim-nox
* vim-nox-py2
Try: sudo apt install <selected package>
nthuos@ubuntu:~$ sudo apt install vim
[sudo] password for nthuos:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
 vim-runtime
Suggested packages:
 ctags vim-doc vim-scripts vim-gnome-py2 | vim-gtk-py2 | vim-gtk3-py2
 | vim-athena-py2 | vim-nox-py2
The following NEW packages will be installed:
 vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 249 not upgraded.
Need to get 6,199 kB of archives.
After this operation, 30.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

type Y to continue

Linux (4/5): install vim

- type `vim`. if installed, you should see this:
 - to exit vim, type `ZZ` (you won't see `ZZ` but you'll be back to bash)



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "nthuos@ubuntu: ~". The window displays the Vim startup message:

```
VIM - Vi IMproved
version 7.4.1689
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

      Help poor children in Uganda!
type :help iccf<Enter>      for information

type :q<Enter>          to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info

0,0-1      All
```

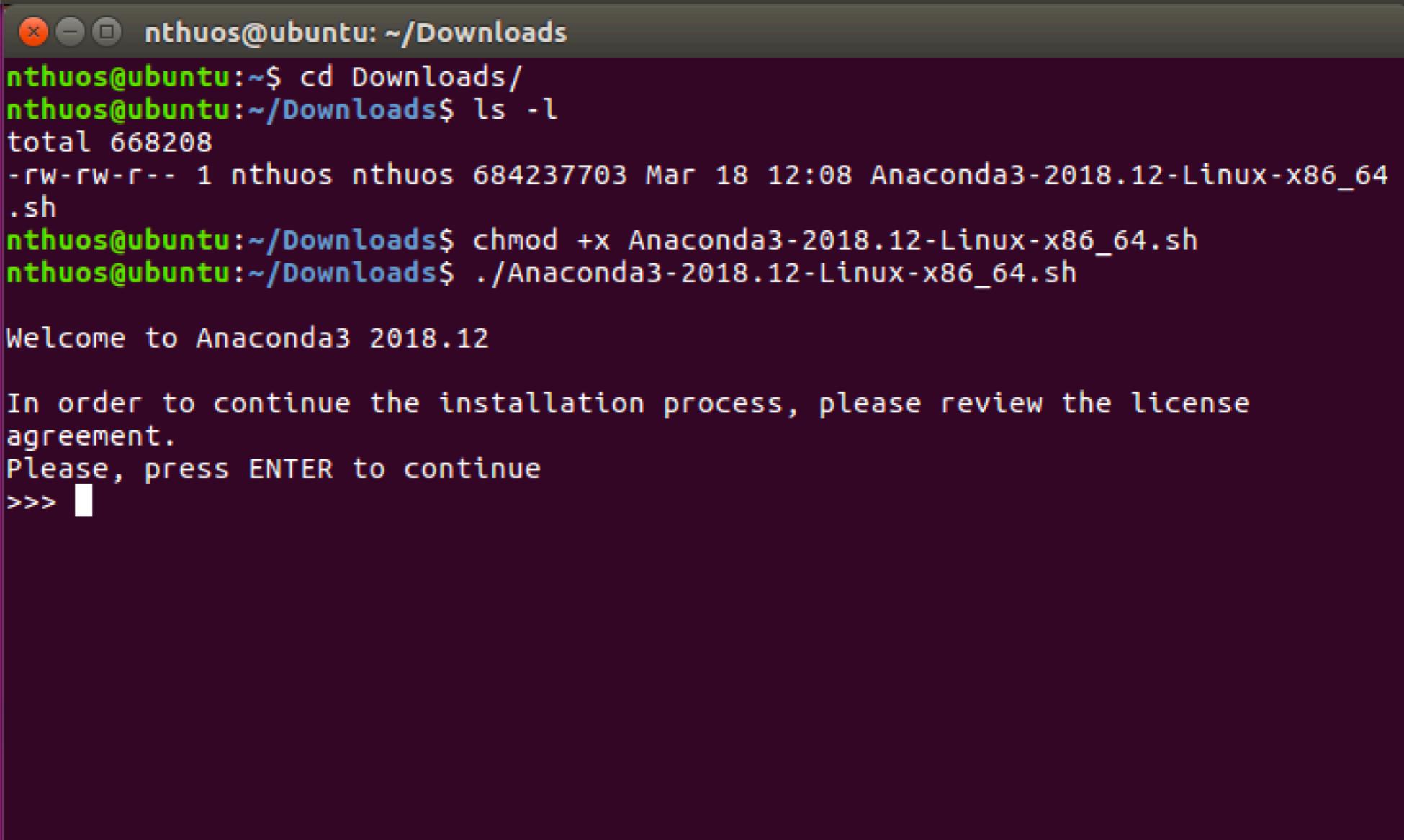
Linux (5/5): upgrade python3

- Ubuntu comes preinstalled with python3
- to upgrade to the latest, type
 - sudo apt-get update
 - sudo apt-get -y upgrade
- To manage python packages,
 - sudo apt-get install -y python3-pip

Installing Jupyter Notebook on Linux

- Browser to download Anaconda Installer
 - <https://www.anaconda.com/distribution/>
 - Default: saves to Downloads folder
- Open a shell
 - \$ cd Downloads
 - \$ chmod +x Anaconda3-2018.12-Linux-x86_64.sh
(or whatever name of the downloaded file is)
 - \$./Anaconda3-2018.12-Linux-x86_64.sh
(and you will be asked several questions during installation)
- Close the shell and reopen a new shell
 - \$ jupyter notebook

Screenshots during installation



A screenshot of a terminal window titled "nthuos@ubuntu: ~/Downloads". The terminal shows the following command sequence:

```
nthuos@ubuntu:~/Downloads
nthuos@ubuntu:~$ cd Downloads/
nthuos@ubuntu:~/Downloads$ ls -l
total 668208
-rw-rw-r-- 1 nthuos nthuos 684237703 Mar 18 12:08 Anaconda3-2018.12-Linux-x86_64
.sh
nthuos@ubuntu:~/Downloads$ chmod +x Anaconda3-2018.12-Linux-x86_64.sh
nthuos@ubuntu:~/Downloads$ ./Anaconda3-2018.12-Linux-x86_64.sh

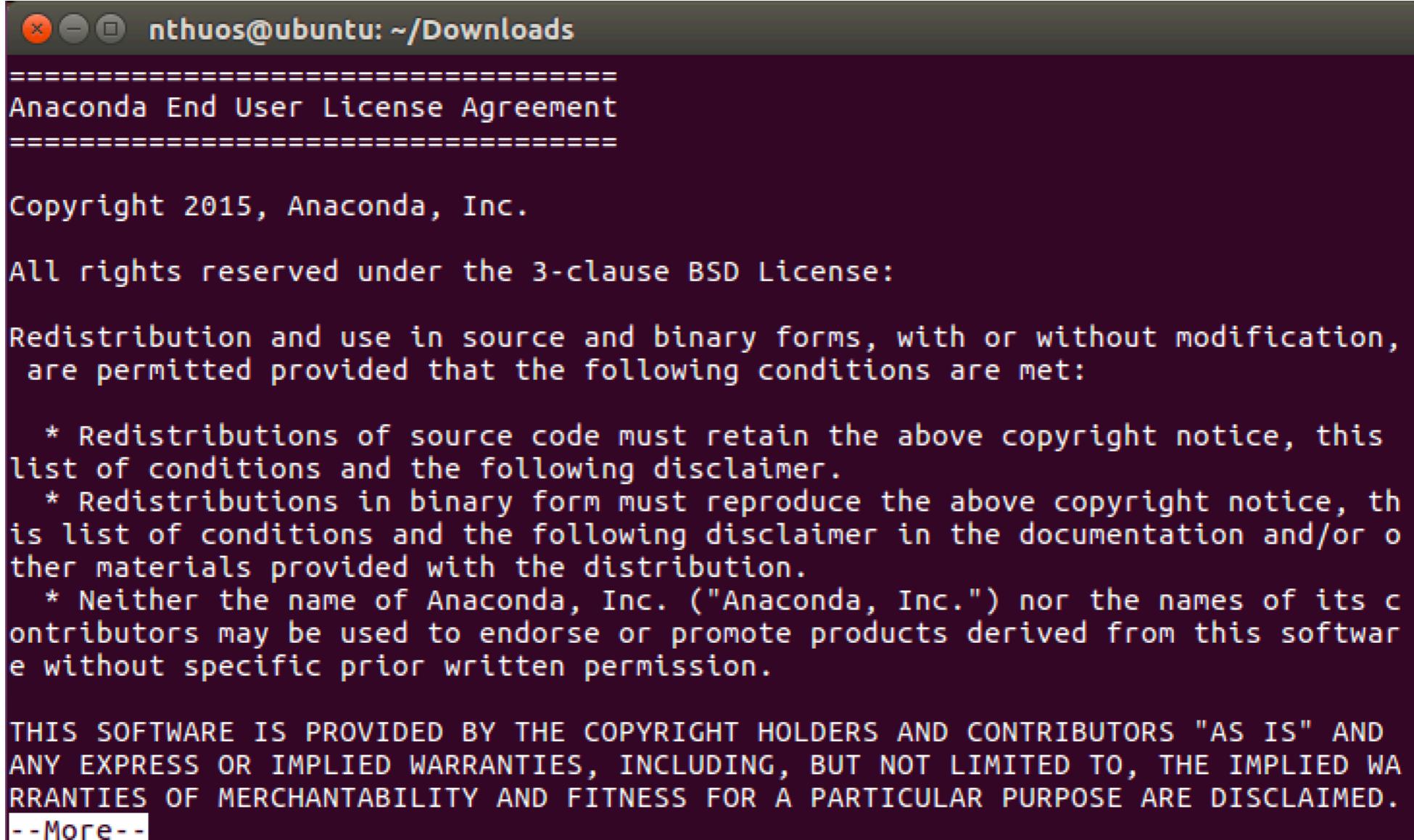
Welcome to Anaconda3 2018.12

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> █
```

The terminal then prompts the user to press Enter to continue, with a small black square cursor placeholder.

type return

Screenshots during installation



nthuos@ubuntu: ~/Downloads

```
=====
Anaconda End User License Agreement
=====

Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

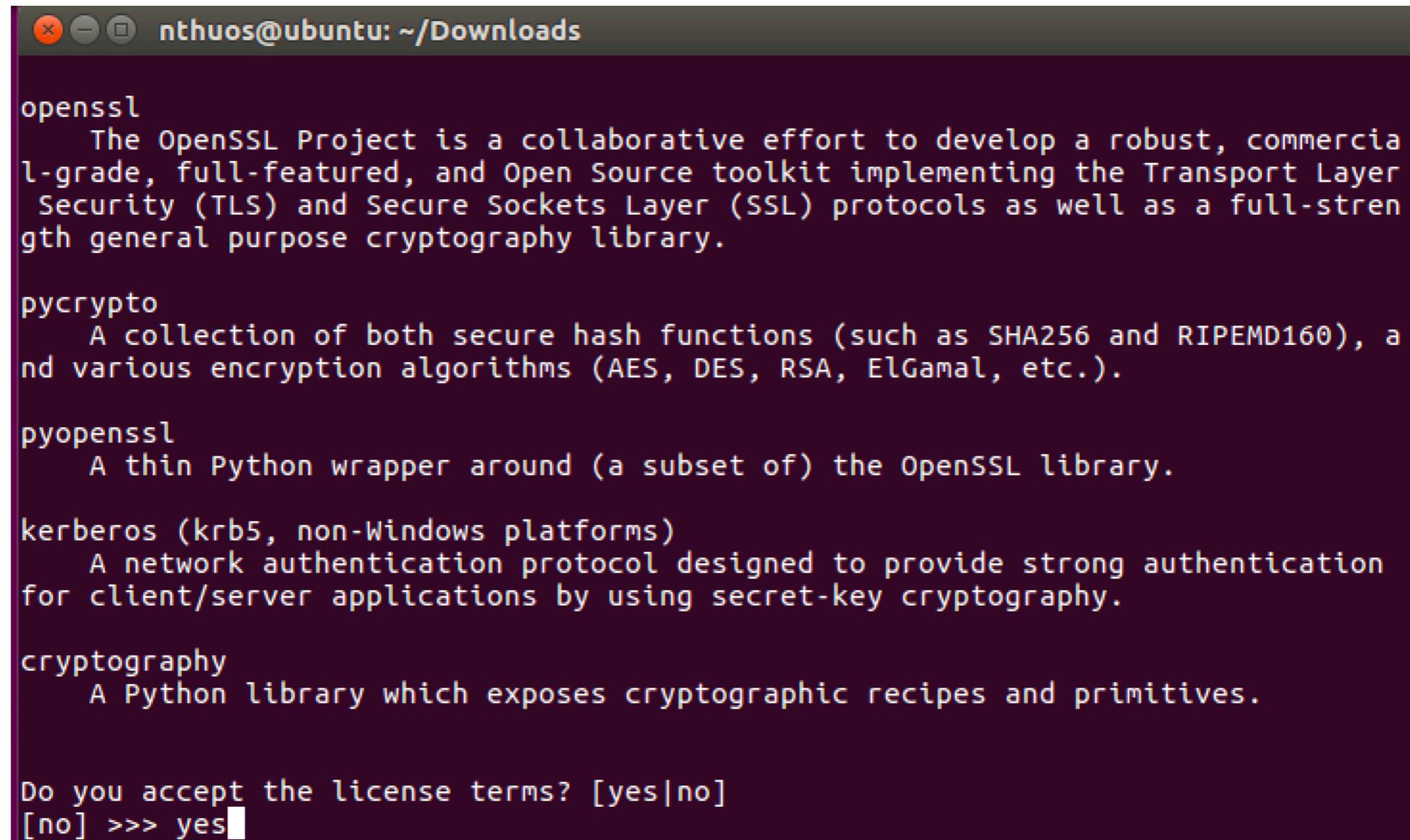
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright notice, th
is list of conditions and the following disclaimer in the documentation and/or o
ther materials provided with the distribution.
    * Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its c
ontributors may be used to endorse or promote products derived from this softwar
e without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WA
RRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
--More--
```

type space

Screenshots during installation



A terminal window titled "nthuos@ubuntu: ~/Downloads" displays the following text:

```
openssl
  The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols as well as a full-strength general purpose cryptography library.

pycrypto
  A collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.).

pyopenssl
  A thin Python wrapper around (a subset of) the OpenSSL library.

kerberos (krb5, non-Windows platforms)
  A network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography.

cryptography
  A Python library which exposes cryptographic recipes and primitives.

Do you accept the license terms? [yes|no]
[no] >>> yes
```

type yes

Screenshots during installation

```
nthuos@ubuntu: ~/Downloads
  nd various encryption algorithms (AES, DES, RSA, ElGamal, etc.).

pyopenssl
  A thin Python wrapper around (a subset of) the OpenSSL library.

kerberos (krb5, non-Windows platforms)
  A network authentication protocol designed to provide strong authentication
  for client/server applications by using secret-key cryptography.

cryptography
  A Python library which exposes cryptographic recipes and primitives.

Do you accept the license terms? [yes|no]
[no] >>> yes

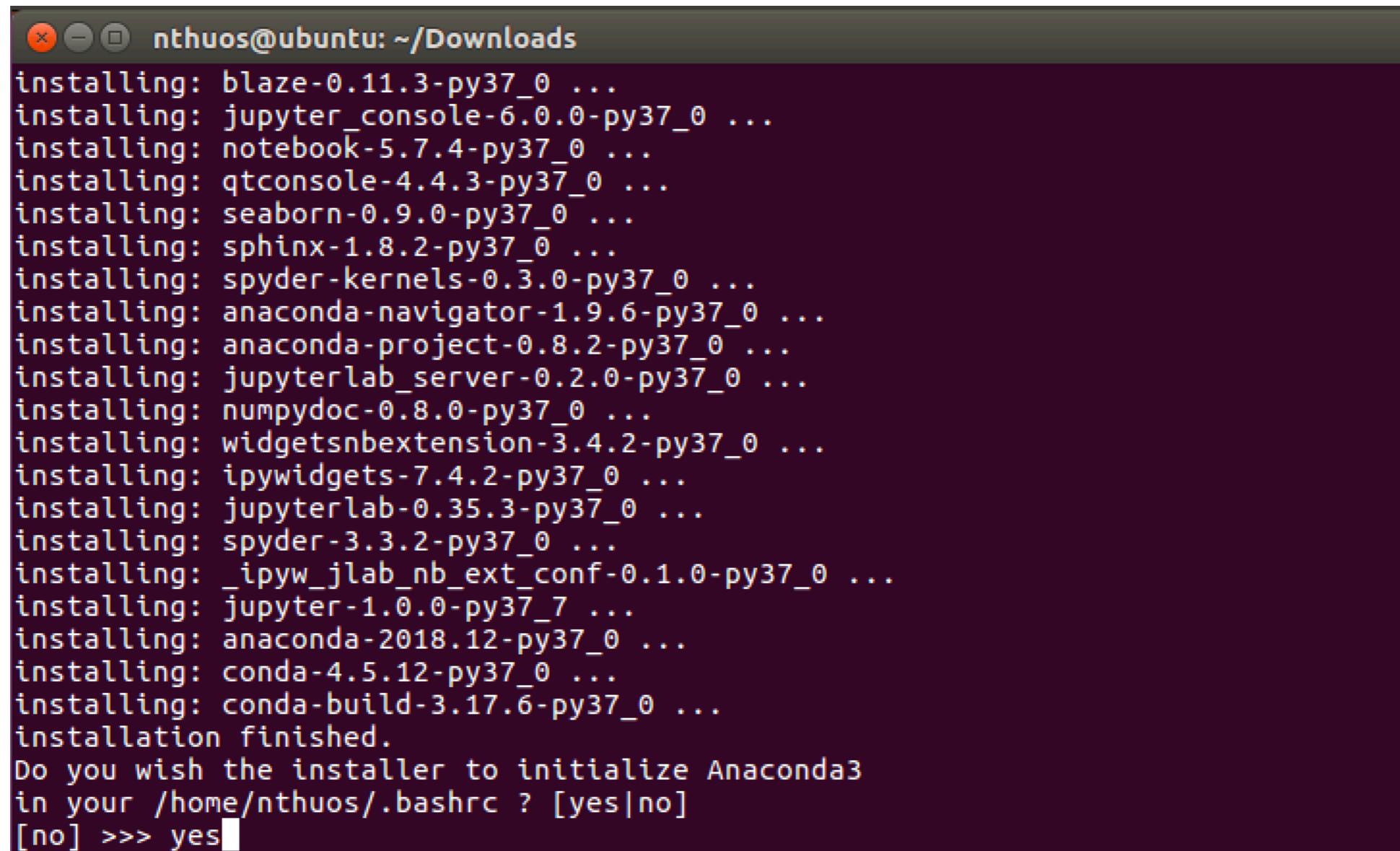
Anaconda3 will now be installed into this location:
/home/nthuos/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/nthuos/anaconda3] >>> 
```

type return

Screenshots during installation

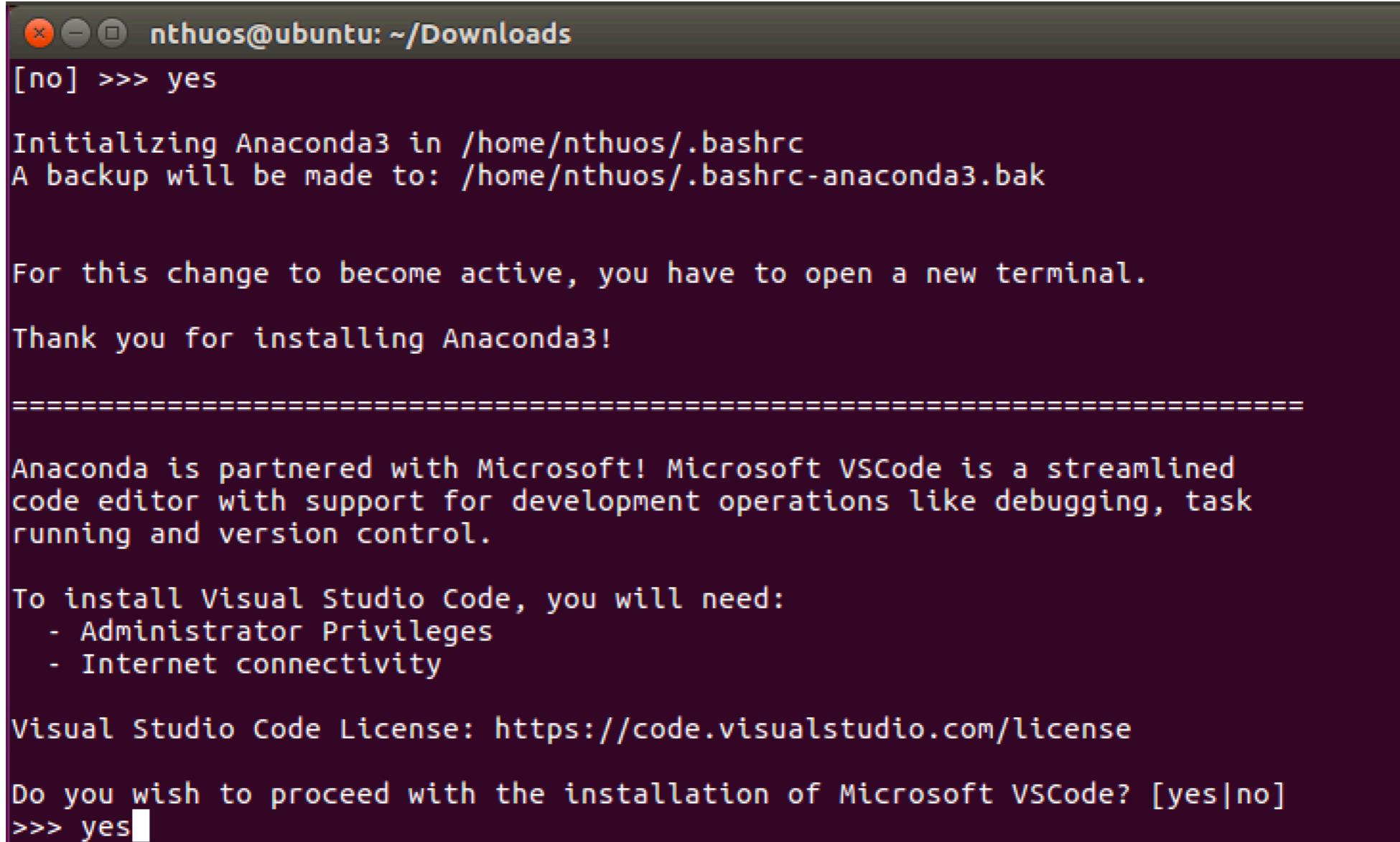


A screenshot of a terminal window titled "nthuos@ubuntu: ~/Downloads". The window displays the output of a package installation process. The text shows multiple "installing" messages for various Python packages, followed by a "installation finished." message and a prompt asking if the installer should initialize Anaconda3 in the user's .bashrc file. The user has typed "yes" at the end of the command line.

```
installing: blaze-0.11.3-py37_0 ...
installing: jupyter_console-6.0.0-py37_0 ...
installing: notebook-5.7.4-py37_0 ...
installing: qtconsole-4.4.3-py37_0 ...
installing: seaborn-0.9.0-py37_0 ...
installing: sphinx-1.8.2-py37_0 ...
installing: spyder-kernels-0.3.0-py37_0 ...
installing: anaconda-navigator-1.9.6-py37_0 ...
installing: anaconda-project-0.8.2-py37_0 ...
installing: jupyterlab_server-0.2.0-py37_0 ...
installing: numpydoc-0.8.0-py37_0 ...
installing: widgetsnbextension-3.4.2-py37_0 ...
installing: ipywidgets-7.4.2-py37_0 ...
installing: jupyterlab-0.35.3-py37_0 ...
installing: spyder-3.3.2-py37_0 ...
installing: _ipyw_jlab_nb_ext_conf-0.1.0-py37_0 ...
installing: jupyter-1.0.0-py37_7 ...
installing: anaconda-2018.12-py37_0 ...
installing: conda-4.5.12-py37_0 ...
installing: conda-build-3.17.6-py37_0 ...
installation finished.
Do you wish the installer to initialize Anaconda3
in your /home/nthuos/.bashrc ? [yes|no]
[no] >>> yes
```

type return

Screenshots during installation



A screenshot of a terminal window titled "nthuos@ubuntu: ~/Downloads". The window contains the following text:

```
[no] >>> yes  
  
Initializing Anaconda3 in /home/nthuos/.bashrc  
A backup will be made to: /home/nthuos/.bashrc-anaconda3.bak  
  
For this change to become active, you have to open a new terminal.  
Thank you for installing Anaconda3!  
  
=====
```

Anaconda is partnered with Microsoft! Microsoft VSCode is a streamlined code editor with support for development operations like debugging, task running and version control.

To install Visual Studio Code, you will need:

- Administrator Privileges
- Internet connectivity

Visual Studio Code License: <https://code.visualstudio.com/license>

Do you wish to proceed with the installation of Microsoft VSCode? [yes|no]
>>> yes

type yes (if you want), and then it will ask you for your password

Command-line Prompt

- Character or string indicating ready for command
 - bash uses \$
 - DOS uses >
 - csh uses %
 - Python uses >>>
- Cursor is to the right of the prompt
- Some prompts may show current directory or user
 - e.g., [MyLaptop:~/Documents/python] me\$ █

First commands

- `pwd` - print [the] working directory
- `ls` - list
- `cd` - change directory
- `mv` - move (or rename)
- `rm` - remove (delete)
- `cp` - copy files
- `mkdir` - make directory
- `man` - manual page for a command

Directory command: pwd

- **pwd** - prints working directory
 - the actual result depends on your own system setup. Suppose user name is "nthuos"
 - Linux example
 - macOS example
 - Cygwin example
 - this maps to C:\cygwin\home\nthuos on Windows

```
$ pwd  
/home/nthuos
```

```
$ pwd  
/Users/nthuos
```

```
$ pwd  
/cygdrive/c/Users/nthuos
```

More ls options

- list files and directory

```
$ ls
```

Desktop	Music	Templates
Documents	Pictures	Videos
Downloads	Public	

- -l option: longer listing for details

```
$ ls -l
```

total 52								
drwxr-xr-x 2 nthuos nthuos 4096 Mar 16 17:25	Desktop							
drwxr-xr-x 2 nthuos nthuos 4096 Sep 9 2018	Documents							
drwxr-xr-x 2 nthuos nthuos 4096 Sep 9 2018	Downloads							
drwxr-xr-x 2 nthuos nthuos 4096 Dec 12 2018	Music							
drwxr-xr-x 2 nthuos nthuos 4096 Feb 3 2019	Pictures							
drwxr-xr-x 2 nthuos nthuos 4096 Mar 12 12:27	Public							
drwxr-xr-x 2 nthuos nthuos 4096 Mar 15 20:23	Templates							
drwxr-xr-x 2 nthuos nthuos 4096 Jan 22 15:23	Videos							

Directory command: ls

- -t option: sorts by last modified date

```
$ ls -lt
total 52
drwxr-xr-x 2 nthuos nthuos 4096 Mar 16 17:25 Desktop
drwxr-xr-x 2 nthuos nthuos 4096 Mar 15 20:23 Templates
drwxr-xr-x 2 nthuos nthuos 4096 Mar 12 12:27 Public
drwxr-xr-x 2 nthuos nthuos 4096 Feb  3  2019 Pictures
drwxr-xr-x 2 nthuos nthuos 4096 Jan 22 15:23 Videos
drwxr-xr-x 2 nthuos nthuos 4096 Dec 12  2018 Music
drwxr-xr-x 2 nthuos nthuos 4096 Sep  9  2018 Documents
drwxr-xr-x 2 nthuos nthuos 4096 Sep  9  2018 Downloads
```

Directory command: ls

- -a option: to show hidden files
 - (names that begin with .)

```
$ ls -a
.
..
.bash_history
.bash_logout
.bashrc
.config
Desktop
Downloads
.gconf
.gnupg
.local
Music
Pictures
.profile
Public
Templates
Videos
.viminfo
```

- . refers to current directory
- .. is directory above

Subdirectories: mkdir

- `mkdir`: make a directory
 - makes a directory (folder) named `src`

```
$ mkdir src
$ ls
Desktop           Music            src
Documents         Pictures          Templates
Downloads         Public           Videos
```

- helps you to organize your files

Subdirectories: cd

- cd: change working directory

```
$ pwd  
/Users/nthuos  
$ cd src  
$ ls  
$ pwd  
/Users/nthuos/src  
$ cd  
$ pwd  
/Users/nthuos
```

- to go one level above, cd ..
- to go back to home directory, just cd

File redirection: >

- if you add `> fileName` after a command, its output goes into the file

```
$ ls
Desktop      Music      src
Documents    Pictures   Templates
Downloads   Public     Videos
$ ls > outFile
$ more outFile
Desktop
Documents
Downloads
Music
Pictures
Public
src
Templates
Videos
```

Moving or renaming a file: mv

- `mv file1 file2`
 - renames *file1* as *file2*
- `mv file(s) dir`
 - move file(s) (or directories) into directory *dir*

```
$ ls
Desktop           Music          Public         Videos
Documents        outFile        src
Downloads        Pictures       Templates
$ mv outFile outFile.txt # renames outFile as outFile.txt
$ mv outFile.txt src      # move outFile.txt into src subdirectory
```

- note: `#` starts a line-terminated comment

Remove a file or directory

- `rm file1 file2 ...`: remove file(s)
 - `rm -i` to interactively confirm each file to remove
- `rmdir dir`: remove directory
 - Note: directory must be empty!
- `rm -r dir` : recursively remove dir
 - This is how to remove a nonempty directory
 - Potentially risky -- use with caution!

Summary: shell commands

- list files
- files: move / rename, remove
- directories: make, change, remove
- path
 - ~ is home directory
 - . is current directory, .. is parent
 - / character is for directory separator

The vim Text Editor

- Purpose
 - edit a plain text file
 - supports syntax highlighting, split screen, mouse cursor positioning (on some terminals)
- Why use vim?
 - work in text mode, no need to switch window
 - modal editor: efficient editing once used to
 - powerful editing commands
- <http://www.study-area.org/tips/vim/index.html>

Vi modal editor

- Command mode
 - keys are interpreted as commands and not shown
 - e.g., moving cursor, cut and paste, save, check
 - : takes line-based commands on bottom of screen
- Insertion mode
 - keys are inserted literally into the text content buffer, rather than interpreted as commands
 - type Escape key (Esc) to get back to command mode

Simple editing with vi

```
$ vi letter.py
```

- edits letter.py if exists, or creates it on save

Entering text in vi

- Type i (not shown) to enter insertion

```
print("hello world") █  
~  
~  
~  
~  
~  
~  
~  
-- INSERT --
```

- Type the Esc key (not e s c letters) to return to command mode. the --INSERT-- disappears.
- backspace, Ctrl-W (backspace word)

Saving file

- Save a file from command mode
 - :w followed by a return

```
print("hello world")
~
~
~
:w █
```

- The status gets updated on the bottom

```
print("hello world")
~
~
~
"letter.py" [New] 1L, 21C written
```

Moving cursor

- In command mode
 - **j** (down), **k** (up), **h** (left), **l** (right)
 - **w** (word forward), **b** (word backward)
 - **Ctrl-F** (forward page), **Ctrl-B** (back page)
 - **Ctrl-E** (scroll up line), **Ctrl-Y** (scroll down line)
 - **^** (beginning of line), **\$** (end of line)
 - **H** (top left corner), **L** (bottom line)
 - **%** (match left/right parentheses or brackets)

Quit

- Save and quit: two ways (command mode)
 - `:wq` (will show on bottom line)
 - `zz` (will not show on screen; no colon)
- Quit without saving
 - `:q!`

Cut, Copy, Paste (all command mode)

- Cut
 - `x` (delete current char)
 - `dw` (delete word), `db` (delete word back)
 - `dd` (delete line), `d2d` (delete 2 lines),
 - `d20d` (delete 20 lines)
 - `d}` (delete till end of paragraph)
 - `D` (delete rest of the line)
- Paste
 - `p` (paste after cursor or line), `P` (paste at cursor or line)
- Copy ("Yank")
 - same as cut, just replace `d` with `y` (or `D` with `Y`)

Different ways to insert

- **a** appends text after current cursor position
- **i** inserts text before current cursor position
- **A** appends text after end of line
- **I** inserts text at beginning of line
- **o** open a new line below current line
- **O** open a line above current line

Syntax highlighting

- from command mode, :syntax on (return)

```
print("hello world")
~
~
~
:syntax on
```

- source code gets highlighted

```
print("hello world")
~
~
~
:syntax on
```

- set line number by :set nu

Now you are ready to write code!

- From bash prompt `$`, you can
 - `$ python3` to enter interactive mode in Python; or
 - Use vim to create a `program.py` file, and
 - `$ python3 program.py` to run the python program in batch mode.