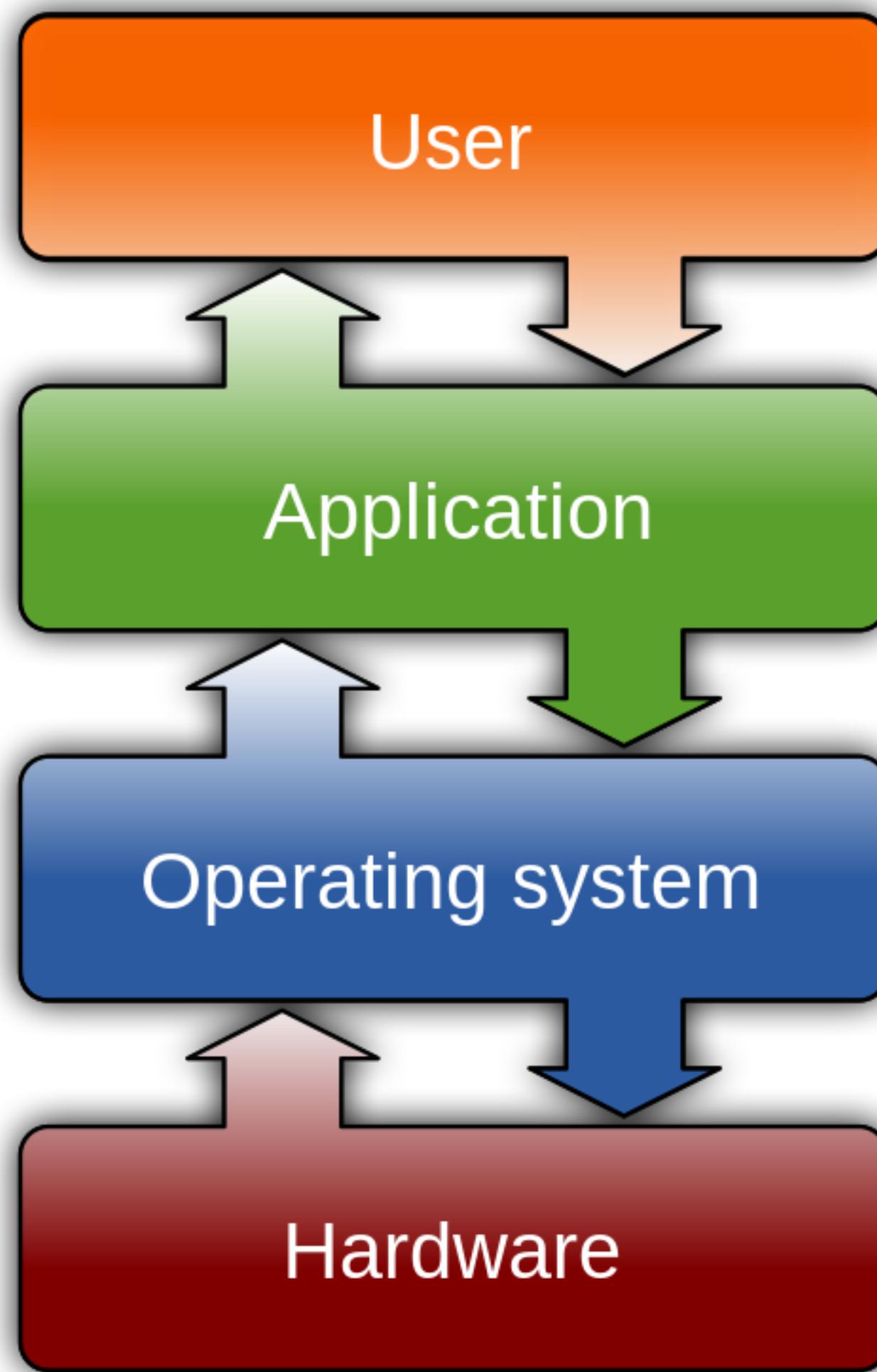


Research Computing Basics

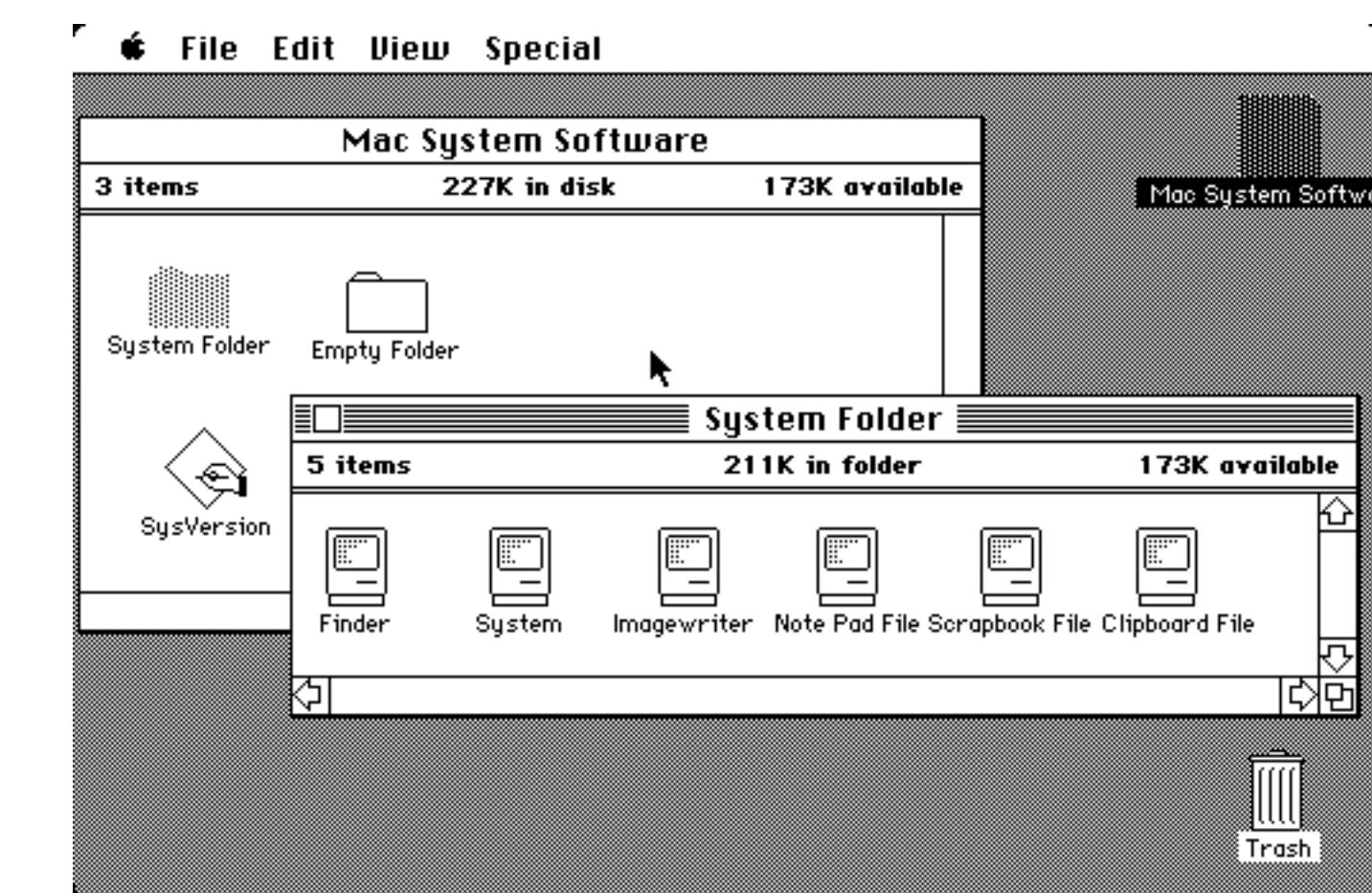
USRP Bootcamp
Chang-Goo Kim



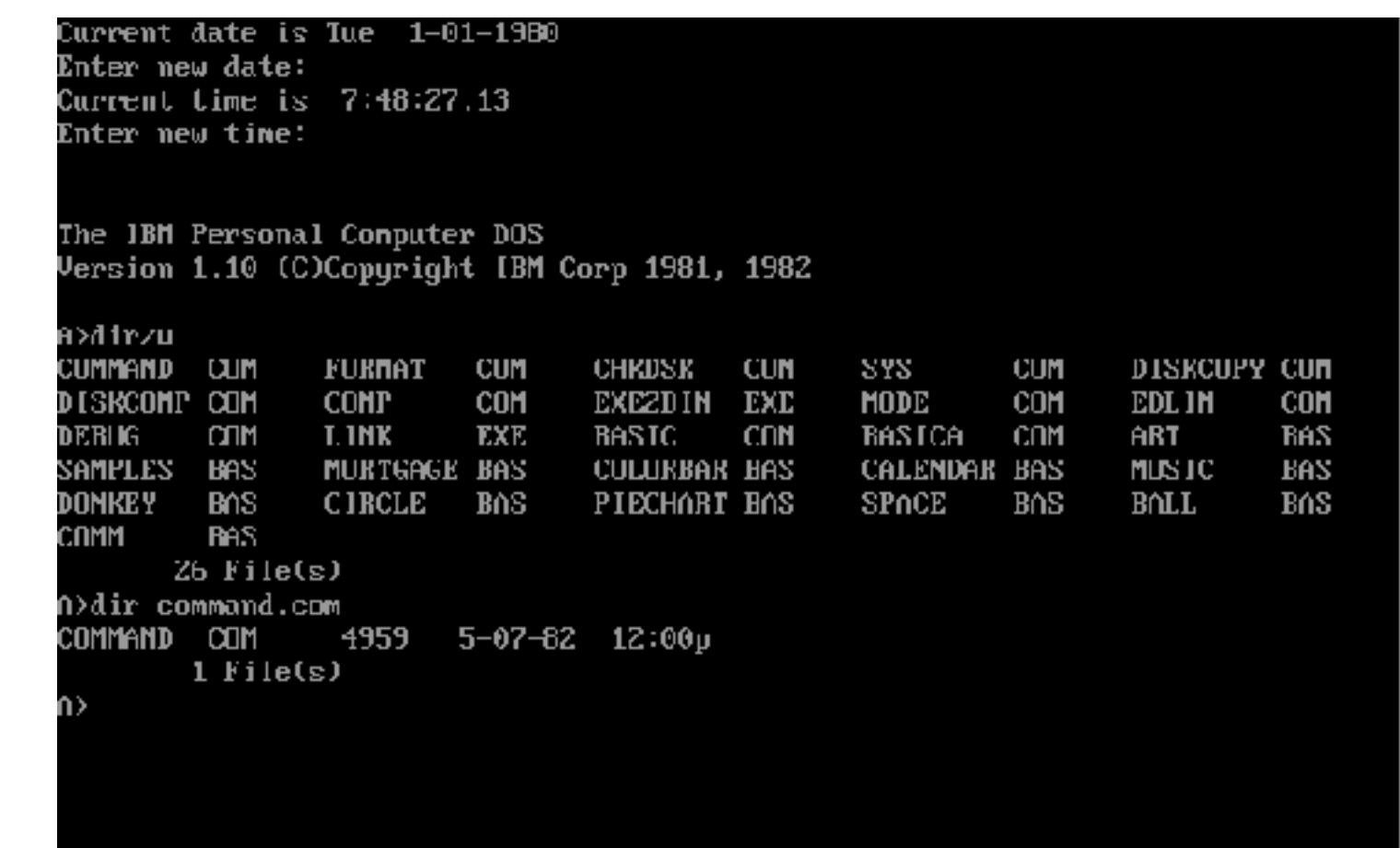
Terminal

**Unix/Linux, macOS, MS Windows
Android's, iOS**

**Personal computers (Desktop, Laptop)
Smartphones
High-performance computers (supercomputers, servers)**



Mac OS – Graphic User Interface



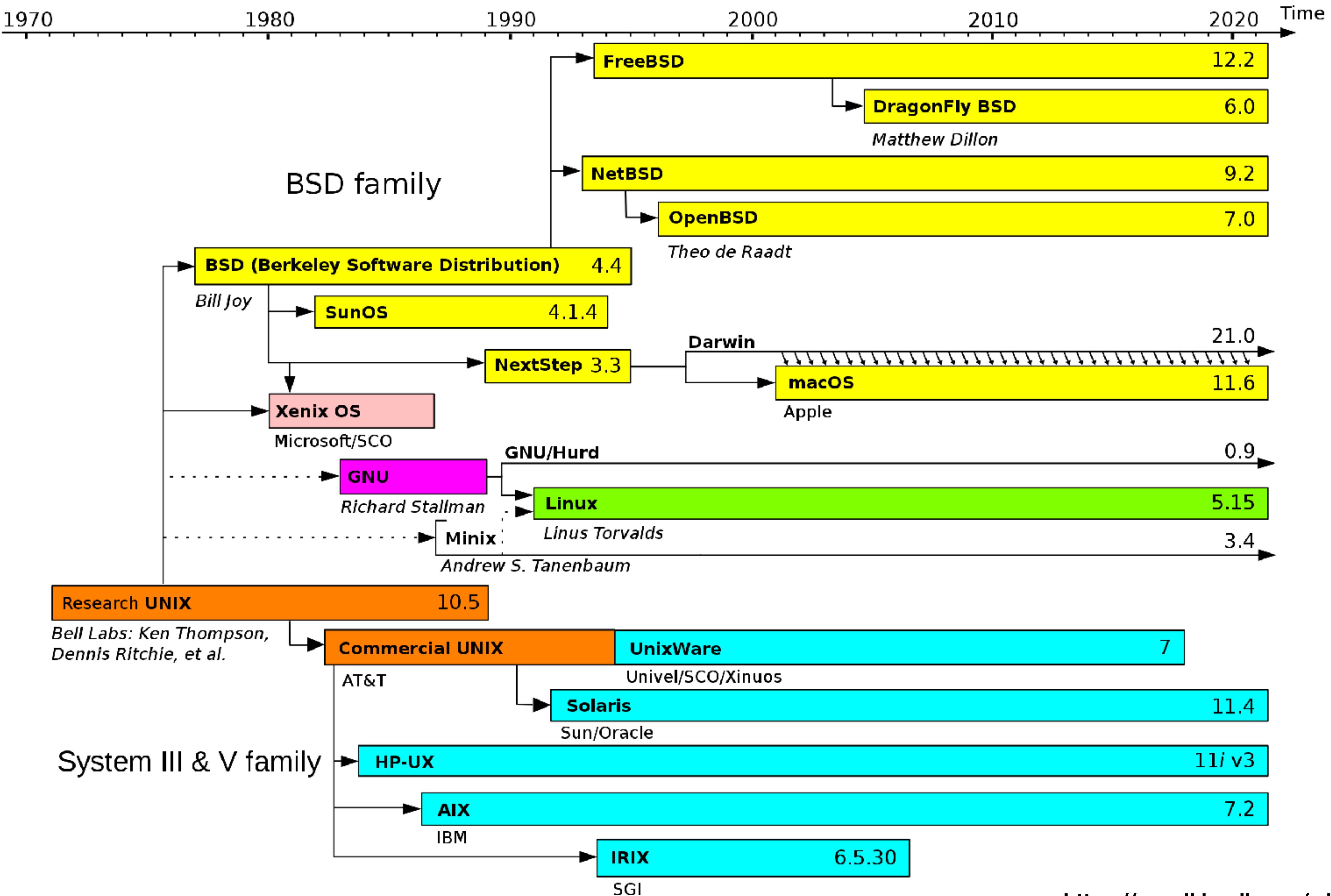
MS-DOS – Command Line Interface

UNIX/Unix-like

- Unix (/ju:niks/; trademarked as UNIX) is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, whose development started in 1969 at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.
- Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties in the late 1970s, leading to a variety of both academic and commercial Unix variants from vendors including University of California, Berkeley (BSD), Microsoft (Xenix), Sun Microsystems (SunOS/Solaris), HP/HPE (HP-UX), and IBM (AIX). In the early 1990s, AT&T sold its rights in Unix to Novell, which then sold the UNIX trademark to The Open Group, an industry consortium founded in 1996. The Open Group allows the use of the mark for certified operating systems that comply with the Single UNIX Specification (SUS).
- Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy". According to this philosophy, the operating system should provide a set of simple tools, each of which performs a limited, well-defined function. A unified and inode-based filesystem (the Unix filesystem) and an inter-process communication mechanism known as "pipes" serve as the main means of communication, and a shell scripting and command language (the Unix shell) is used to combine the tools to perform complex workflows.
- Unix distinguishes itself from its predecessors as the first portable operating system: almost the entire operating system is written in the C programming language, which allows Unix to operate on numerous platforms.
- A Unix-like (sometimes referred to as UN*X or *nix) operating system is one that behaves in a manner similar to a Unix system, although not necessarily conforming to or being certified to any version of the Single UNIX Specification. A Unix-like application is one that behaves like the corresponding Unix command or shell. There is no technical standard defining the term, and opinions can differ about whether or the degree to which a particular operating system or application is Unix-like.

<https://en.wikipedia.org/wiki/Unix>

<https://en.wikipedia.org/wiki/Unix-like>



Linux/Darwin/macOS

- The Linux kernel is a free and open-source, monolithic, modular, multitasking, Unix-like operating system kernel. It was originally authored in 1991 by Linus Torvalds for his i386-based PC, and it was soon adopted as the kernel for the GNU operating system, which was written to be a free (*libre*) replacement for UNIX.
- Linux is a family of open-source Unix-like operating systems based on the Linux kernel.
- Darwin is an open-source Unix-like operating system first released by Apple Inc. in 2000. It is composed of code derived from NeXTSTEP, BSD, Mach, and other free software projects' code, as well as code developed by Apple.
- Darwin forms the Unix-based core set of components upon which macOS (previously OS X and Mac OS X), iOS, watchOS, tvOS, iPadOS and bridgeOS are based.

<https://en.wikipedia.org/wiki/Linux>

[https://en.wikipedia.org/wiki/Darwin_\(operating_system\)](https://en.wikipedia.org/wiki/Darwin_(operating_system))

Bourne shell/Bash

- In computing, a shell is a computer program which exposes an operating system's services to a human user or other programs. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. It is named a shell because it is the outermost layer around the operating system.
- The Bourne shell (sh) is a shell command-line interpreter for computer operating systems. The Bourne shell was the default shell for Version 7 Unix. Unix-like systems continue to have /bin/sh—which will be the Bourne shell, or a symbolic link or hard link to a compatible shell—even when other shells are used by most users.
- Bash (Bourne Again Shell) is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell.
- Bash is a command processor that typically runs in a text window where the user types commands that cause actions. Bash can also read and execute commands from a file, called a shell script. Like most Unix shells, it supports filename globbing (wildcard matching), piping, here documents, command substitution, variables, and control structures for condition-testing and iteration.

[https://en.wikipedia.org/wiki/Shell_\(computing\)](https://en.wikipedia.org/wiki/Shell_(computing))

https://en.wikipedia.org/wiki/Bourne_shell

Terminal



- A terminal emulator, or terminal application, or terminal window, is a computer program that emulates a video terminal within some other display architecture. Though typically synonymous with a shell or text terminal, the term terminal covers all remote terminals, including graphical interfaces. A terminal emulator inside a graphical user interface is often called a terminal window.
- A terminal window allows the user access to a text terminal and all its applications such as command-line interfaces (CLI) and text user interface (TUI) applications. These may be running either on the same machine or on a different one via telnet, ssh, dial-up, or over a direct serial connection. On Unix-like operating systems, it is common to have one or more terminal windows connected to the local machine.

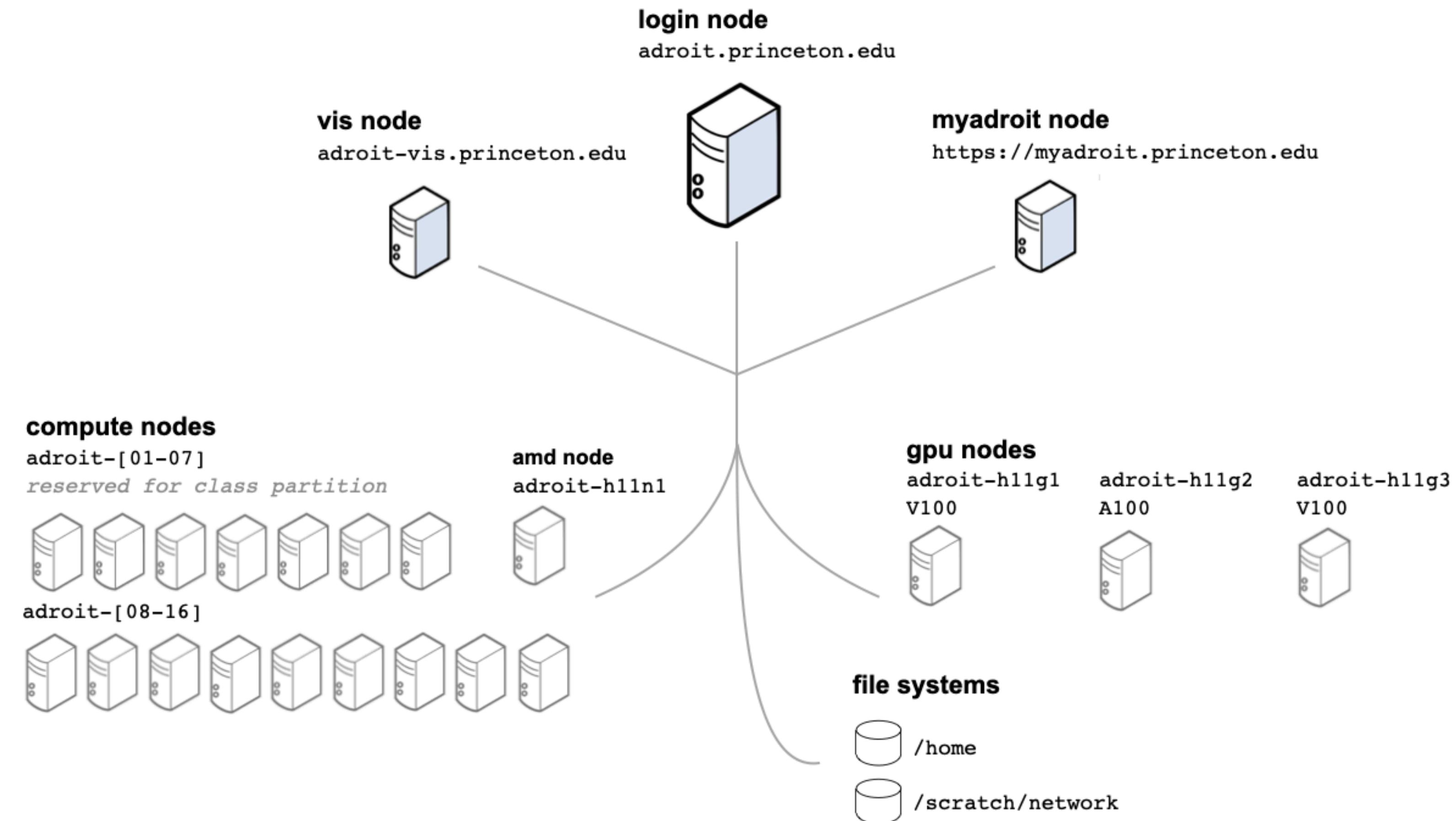
Basic shell commands

- ls — List directory contents
- echo — Prints text to the terminal window
- touch — Creates a file
- mkdir — Create a directory
- grep — search
- man — Print manual or get help for a command
- pwd — Print working directory
- cd — Change directory
- mv — Move or rename directory
- rmdir — Remove directory (rm -d)
- locate — Locate a specific file or directory
- less — view the contents of a text file
- > — redirect stdout
- cat — Read a file, create a file, and concatenate files
- | — Pipe
- head — Read the start of a file
- tail — Read the end of a file
- chmod — Sets the file permissions flag on a file or folder
- history — list your most recent commands
- clear — Clear your terminal window
- cp — copy files and directories
- top - display Linux processes

<https://www.educative.io/blog/bash-shell-command-cheat-sheet>

<https://github.com/gabeclass/introcmdline>

adroit



adroit (cont'd)

- If you would like an account on Adroit, please fill out the Adroit Registration form to request an account.
 - <https://forms.rc.princeton.edu/registration/>
- Once you have been granted access to Adroit, you can connect by opening an SSH client and typing the following SSH command (VPN required from off-campus)

```
$ ssh <YourNetID>@adroit.princeton.edu
```

- If you prefer to navigate Adroit through a graphical user interface rather than the Linux command line, Adroit has a web portal option called MyAdroit (VPN required from off-campus)
 - myadroit.princeton.edu
- SSH: https://princeton.service-now.com/service?id=kb_article&sys_id=f52a27064f9ca20018ddd48e5210c72d
- VPN: https://princeton.service-now.com/service?id=kb_article&sys_id=6023
- GlobalProtect VPN: https://princeton.service-now.com/service?id=kb_article&sys_id=KB0012373
- https://github.com/PrincetonUniversity/removing_tedium

myadroit.princeton.edu

Dashboard - MyAdroit - Adroit cluster

MyAdroit - Adroit cluster Files ▾ Jobs ▾ Clusters ▾ Interactive Apps ▾

PRINCETON UNIVERSITY Welcome to MyAdroit - web interface to Adroit cluster

powered by OPEN OnDemand

Display a menu

OnDemand version: v2.0.20

myadroit.princeton.edu

Dashboard - MyAdroit - Adroit cluster

MyAdroit - Adroit cluster Files ▾ Jobs ▾ Clusters ▾ Interactive Apps ▾

PRIN
UNIVERSITY

>_Adroit Cluster Shell Access
>_Adroit Visualization node Shell Access

web interface to Adroit cluster

powered by
OPEN **OnDemand**

Display a menu for "https://myadroit.princeton.edu/pun/sys/shell/ssh/adroit"

OnDemand version: v2.0.20

The screenshot shows a macOS terminal window with a dark theme. The title bar indicates the session is connected to 'myadroit.princeton.edu'. The user is running a command-line interface on a host named 'adroit'. The session starts with a 'last login' message, followed by several 'ls' commands listing files in the current directory. The user navigates to the directory '/usrp-sciprog' and lists its contents. The terminal window includes standard macOS window controls (red, yellow, green buttons) and a menu bar with various application icons.

```
Last login: Mon May 16 18:53:51 2022 from 172.21.2.7
[changgoo@adroit5 ~]$ ls
Fall_Bootcamp_2019  intel  Intro_to_Perf_Vectorization  ondemand  usrp-sciprog
[changgoo@adroit5 ~]$ pwd
/home/changgoo
[changgoo@adroit5 ~]$ ls -alh
total 144K
drwx-----. 13 changgoo pustaff 4.0K May 16 19:03 .
drwxr-xr-x. 2356 root      root   56K May 12 14:24 ..
-rw-----.  1 changgoo pustaff 2.4K Nov  1 2019 .bash_history
-rw-r--r--.  1 changgoo pustaff 18 Oct 30 2019 .bash_logout
-rw-r--r--.  1 changgoo pustaff 193 Oct 30 2019 .bash_profile
-rw-r--r--.  1 changgoo pustaff 231 Oct 30 2019 .bashrc
drwxr-xr-x.  3 changgoo pustaff 32 Oct 31 2019 .cache
drwx-----.  3 changgoo pustaff 33 Oct 31 2019 .dbus
-rw-r--r--.  1 changgoo pustaff 334 Oct 30 2019 .emacs
drwxr-xr-x.  5 changgoo pustaff 62 Oct 30 2019 Fall_Bootcamp_2019
drwxr-xr-x.  3 changgoo pustaff 28 Oct 31 2019 .intel
drwxr-xr-x.  3 changgoo pustaff 44 Oct 31 2019 intel
drwxr-xr-x.  4 changgoo pustaff 4.0K Oct 31 2019 Intro_to_Perf_Vectorization
-rw-r--r--.  1 changgoo pustaff 172 Oct 30 2019 .kshrc
drwxr-xr-x.  4 changgoo pustaff 51 Oct 30 2019 .mozilla
drwxr-xr-x.  3 changgoo pustaff 26 May  4 12:57 ondemand
drwxr----.  3 changgoo pustaff 27 Oct 31 2019 .pki
drwx-----.  2 changgoo pustaff 77 May 16 18:58 .ssh
drwxr-xr-x.  8 changgoo pustaff 268 May 16 19:03 usrp-sciprog
-rw-----.  1 changgoo pustaff 6.9K Nov  1 2019 .viminfo
-rw-----.  1 changgoo pustaff 159 Nov  1 2019 .Xauthority
-rw-r--r--.  1 changgoo pustaff 658 Oct 30 2019 .zshrc
[changgoo@adroit5 ~]$ cd usrp-sciprog/
[changgoo@adroit5 usrp-sciprog]$ ls
activities.md  bios  data.txt  day1  day2  day3  day4  LICENSE  opt-out.md  README.md  SETUP-INSTALLATION.md
[changgoo@adroit5 usrp-sciprog]$ pwd
/home/changgoo/usrp-sciprog
[changgoo@adroit5 usrp-sciprog]$
```

Display a menu

Access Control List

```
drwxr-xr-x.    8 changgoo pustaff 268 May 16 19:03 usrp-sciprog
```

- The first 10 symbols denote the Access Control List (ACL) which refers to
 - the type of file (- for file, d for directories),
 - the rights for reading (r), writing (w) and executing (x) for the current user (rwx)
 - the rights for reading (r), writing (w) and executing (x) for the members of the group pustaff (r-x)
 - the rights for reading (r), writing (w) and executing (x) for the other users (r-x).
- It is possible to change these rights. This is done with the [chmod](#) command:

The screenshot shows a terminal window titled "Dashboard - MyAdroit - Adroit cluster". The terminal session is on host "adroit" and shows the following command sequence:

```
[changgoo@adroit5 usrp-sciprog]$ touch test_file
[changgoo@adroit5 usrp-sciprog]$ ls -alh test_file
-rw-r--r--. 1 changgoo pustaff 0 May 16 19:15 test_file
[changgoo@adroit5 usrp-sciprog]$ chmod 600 test_file
[changgoo@adroit5 usrp-sciprog]$ ls -alh test_file
-rw-----. 1 changgoo pustaff 0 May 16 19:15 test_file
[changgoo@adroit5 usrp-sciprog]$ rm test_file
[changgoo@adroit5 usrp-sciprog]$ ls -alh
total 32K
drwxr-xr-x. 8 changgoo pustaff 268 May 16 19:16 .
drwx-----. 13 changgoo pustaff 4.0K May 16 19:03 ..
-rw-r--r--. 1 changgoo pustaff 2.2K May 16 19:03 activities.md
drwxr-xr-x. 2 changgoo pustaff 35 May 16 19:03 bios
-rw-r--r--. 1 changgoo pustaff 640 May 16 19:03 data.txt
drwxr-xr-x. 2 changgoo pustaff 282 May 16 19:03 day1
drwxr-xr-x. 3 changgoo pustaff 155 May 16 19:03 day2
drwxr-xr-x. 3 changgoo pustaff 160 May 16 19:03 day3
drwxr-xr-x. 5 changgoo pustaff 122 May 16 19:03 day4
drwxr-xr-x. 8 changgoo pustaff 211 May 16 19:03 .git
-rw-r--r--. 1 changgoo pustaff 107 May 16 19:03 .gitignore
-rw-r--r--. 1 changgoo pustaff 1.2K May 16 19:03 LICENSE
-rw-r--r--. 1 changgoo pustaff 2.5K May 16 19:03 opt-out.md
-rw-r--r--. 1 changgoo pustaff 3.6K May 16 19:03 README.md
-rw-r--r--. 1 changgoo pustaff 3.7K May 16 19:03 SETUP-INSTALLATION.md
[changgoo@adroit5 usrp-sciprog]$
```

The terminal window has a dark theme and includes standard macOS-style window controls at the top.

The three numbers in the arguments of chmod refer to the right of the owner, group, and all users, respectively.

number between 0 and seven will edit the rights based on their binary version.

Exercise) use chmod to give everyone the reading and writing permission (rw- = 110 = ?)

Display a menu

The screenshot shows a macOS terminal window with the following details:

- Address Bar:** myadroit.princeton.edu
- User Information:** changgoo@adroit5:~/usrp-sciprog
- Host:** adroit
- Themes:** Default

The terminal session output is as follows:

```
[changgoo@adroit5 usrp-sciprog]$ mkdir test_directory
[changgoo@adroit5 usrp-sciprog]$ ls -alh
total 36K
drwxr-xr-x.  9 changgoo pustaff  4.0K May 16 19:26 .
drwxr-xr-x. 13 changgoo pustaff  4.0K May 16 19:03 ..
-rw-r--r--.  1 changgoo pustaff 2.2K May 16 19:03 activities.md
drwxr-xr-x.  2 chang goo pustaff   35 May 16 19:03 bios
-rw-r--r--.  1 chang goo pustaff 640 May 16 19:03 data.txt
drwxr-xr-x.  2 chang goo pustaff 282 May 16 19:03 day1
drwxr-xr-x.  3 chang goo pustaff 155 May 16 19:03 day2
drwxr-xr-x.  3 chang goo pustaff 160 May 16 19:03 day3
drwxr-xr-x.  5 chang goo pustaff 122 May 16 19:03 day4
drwxr-xr-x.  8 chang goo pustaff 211 May 16 19:03 .git
-rw-r--r--.  1 chang goo pustaff 107 May 16 19:03 .gitignore
-rw-r--r--.  1 chang goo pustaff 1.2K May 16 19:03 LICENSE
-rw-r--r--.  1 chang goo pustaff 2.5K May 16 19:03 opt-out.md
-rw-r--r--.  1 chang goo pustaff 3.6K May 16 19:03 README.md
-rw-r--r--.  1 chang goo pustaff 3.7K May 16 19:03 SETUP-INSTALLATION.md
drwxr-xr-x.  2 chang goo pustaff 10 May 16 19:26 test_directory
[changgoo@adroit5 usrp-sciprog]$ cd test_directory/
[changgoo@adroit5 test_directory]$ ls
[changgoo@adroit5 test_directory]$ ls -alh
total 4.0K
drwxr-xr-x.  2 chang goo pustaff 10 May 16 19:26 .
drwxr-xr-x.  9 chang goo pustaff 4.0K May 16 19:26 ..
[changgoo@adroit5 test_directory]$ cd ..
[changgoo@adroit5 usrp-sciprog]$ ls
activities.md bios data.txt day1 day2 day3 day4 LICENSE opt-out.md README.md SETUP-INSTALLATION.md test_directory
[changgoo@adroit5 usrp-sciprog]$ rm test_directory/
rm: cannot remove 'test_directory/': Is a directory
[changgoo@adroit5 usrp-sciprog]$
```

Display a menu

The screenshot shows a web browser window with a dark theme, displaying a terminal session on a host named 'adroit'. The terminal window has a title bar 'Dashboard - MyAdroit - Adroit cluster' and a user indicator 'changgoo@adroit5:~/usrp-sciprog'. The content of the terminal is the .bashrc file from the user's home directory. The file includes global definitions sourcing /etc/bashrc, comments about systemctl auto-paging, and user-specific aliases/functions. The terminal prompt ends with a red cursor at the end of the line.

```
Host: adroit
[changgoo@adroit5 usrp-sciprog]$ cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
[changgoo@adroit5 usrp-sciprog]$ less ~/.bashrc
[changgoo@adroit5 usrp-sciprog]$ more ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
[changgoo@adroit5 usrp-sciprog]$ tail ~/.bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
[changgoo@adroit5 usrp-sciprog]$
```

Exercise

- clear terminal; move to your home directory; print working directory
- create a folder “exercise1”
- change directory to “exercise1”; print working directory
- create files “file1” ... “file10”
- list files in the long format and redirect it to “file_list”
- print the contents of “file_list” to terminal
- change the file permission to give everyone read and write access
- move to home directory; remove “exercise1”

```
$ clear
```

```
$ cd
```

```
$ pwd
```

```
$ mkdir exercise1
```

```
$ cd exercise1/
```

```
$ pwd
```

```
$ touch file{1..10}
```

```
$ ls -l > file_list
```

```
$ cat file_list
```

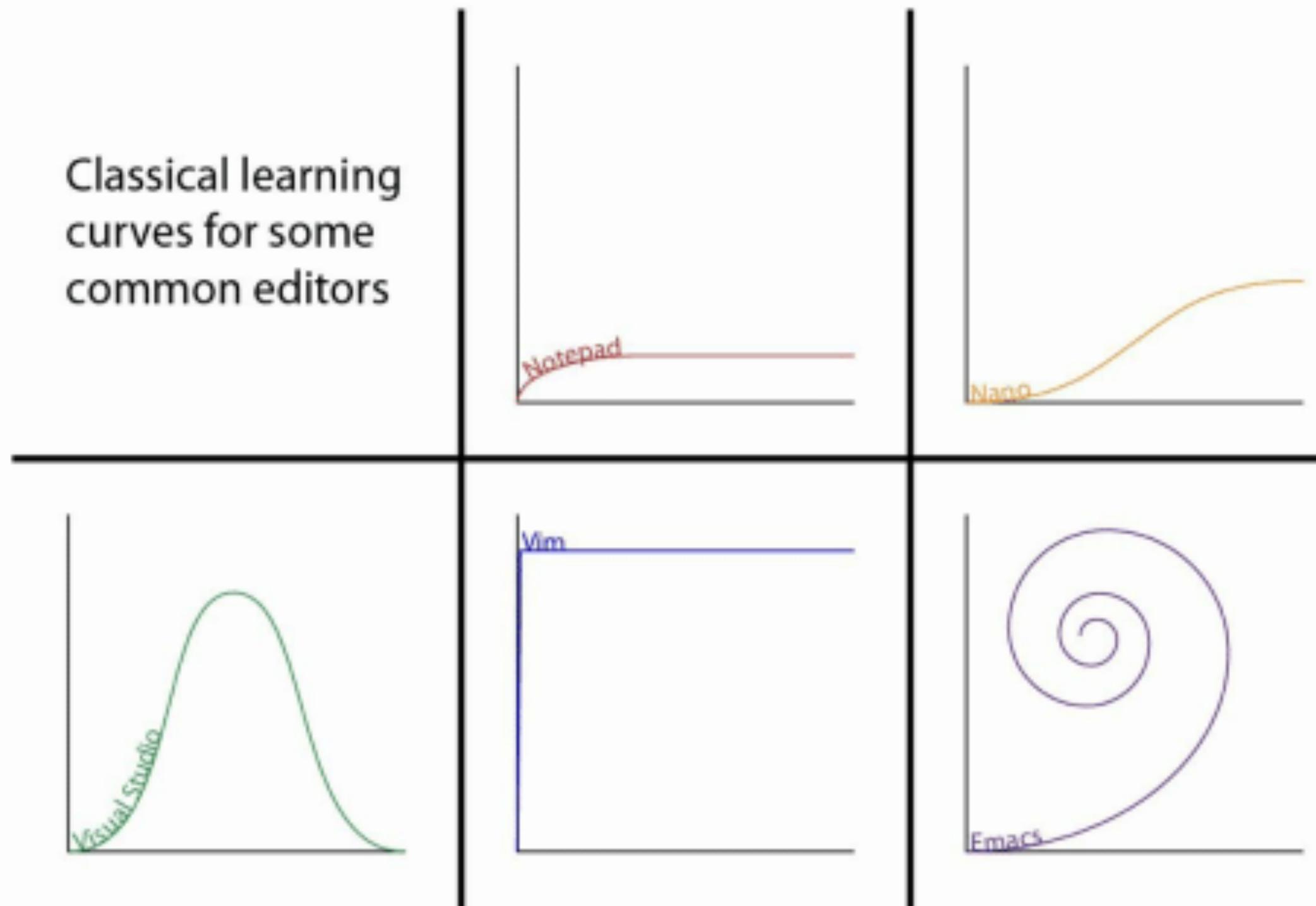
```
$ chmod 666 *
```

```
$ ls -l
```

```
$ cd
```

```
$ rm -rd exercise1/
```

Editor: vim (or emacs)



If you don't use VIM why?



The screenshot shows a terminal window with a dark theme. The title bar reads "Dashboard - MyAdroit - Adroit cluster". The user is connected to "Host: adroit" and the session ID is "changgoo@adroit5:~/usrp-sciprog". The terminal window displays the "General Commands Manual" for "VIM(1)".

VIM(1)

NAME
vim - Vi IMproved, a programmer's text editor

SYNOPSIS

```
vim [options] [file ..]
vim [options] -
vim [options] -t tag
vim [options] -q [errorfile]

ex gex
view
gvim gview vimx evim eview
rvim rview rgvim rgview
```

DESCRIPTION

Vim is a text editor that is upwards compatible to Vi. It can be used to edit all kinds of plain text. It is especially useful for editing programs.

Display a menu

A screenshot of a terminal window titled "Dashboard - MyAdroit - Adroit cluster". The window shows the Vim Tutor version 1.7. The text in the terminal is as follows:

```
Host: adroit
=====
= Welcome to the VIM Tutor - Version 1.7 =
=====

Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this. This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text. Make a copy of this
file to practice on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use. That means that you need to execute the commands to learn them
properly. If you only read the text, you will forget the commands!

Now, make sure that your Caps-Lock key is NOT depressed and press
the j key enough times to move the cursor so that Lesson 1.1
completely fills the screen.

~~~~~
```

Lesson 1.1: MOVING THE CURSOR

```
** To move the cursor, press the h,j,k,l keys as indicated. **
^
k           Hint: The h key is at the left and moves left.
< h      l >       The l key is at the right and moves right.
j           The j key looks like a down arrow.
v
```

1. Move the cursor around the screen until you are comfortable.
2. Hold down the down key (j) until it repeats.
Now you know how to move to the next lesson.
3. Using the down key, move to Lesson 1.2.

Display a menu G6gurd" 970 lines, 33247 characters

Okay, here is VS code

The screenshot shows a VS Code workspace titled "dump_zprof.c - Untitled [Workspace]". The interface includes:

- EXPLORER**: Shows a tree view of files and folders, including "GROUP 1" and "GROUP 2" under "OPEN EDITORS", and "UNTITLED (WORKSPACE)" containing C/C++ source files like "ath_array.c", "ath_log.c", and "config.h".
- OPEN EDITORS**: Displays several code files: "load_sim.py", "dump_zprof.c", "read_zprof.py", "zprof.py", "phase.py", and "load_sim_tigress_ncr.py".
- PROBLEMS**: A list of errors and warnings from the build process.
- OUTPUT**: A terminal window showing the build log and command-line interactions.
- DEBUG CONSOLE**: A terminal window showing the output of the debugger.
- TERMINAL**: A terminal window showing the command-line history and current session.
- PAGES**: A sidebar with links to "bash script" and "bash TUI".
- CODE**: The main editor area showing the "zprof.py" file content:

```
# zprof.py
import es
import es.path as osp
import glob
import xarray as xr
import numpy as np
import astropy.units as au
import astropy.constants as ac
import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm, Normalize

from ..load_sim import LoadSim
from ..io.read_zprof import read_zprof_all, ReadZprofBase
from ..classic.utils import texteffect

class Zprof(ReadZprofBase):

    def read_zprof_new(self, phase='all', savedir=None, force_override=False):
        f = self.files['zprof'][0]
        nphase = len(glob.glob(f[:f.rfind('phase')]+'*.zprof'))

        if phase == 'all':
            phase = ['phase{:d}'.format(i) for i in range(1,nphase)]
        else:
            phase = np.atleast_1d(phase)

        zplist = []
        for ph in phase:
            zp = self._read_zprof(phase=ph, savedir=savedir,
                                  force_override=force_override)
            zplist.append(zp.expand_dims('phase').assign_coords(phase=[ph]))
        zp=xr.concat(zplist,dim='phase')

        self.zp = zp
```

The terminal window at the bottom shows a series of commands being run, including "qsub", "qstat", and "qdel" to manage batch jobs, and "sbatch" to submit new ones. It also shows the user navigating through the codebase and running scripts like "tigress_tutorial.ipynb".

Exercise

```
$ mkdir ~/usrp_working
```

```
$ vi ~/.bashrc
```

- add user-specific aliases at the end of the file
- alias cdwork="cd ~/usrp_working/"
- save and exit

```
$ cdwork
```

```
$ source ~/.bash_profile
```

```
$ cdwork
```

```
$ pwd
```

SSH/PuTTY

- The Secure Shell Protocol (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Its most notable applications are remote login and command-line execution.
- SSH was designed on Unix-like operating systems, as a replacement for Telnet and for unsecured remote Unix shell protocols.
- SSH was first designed in 1995 by Finnish computer scientist Tatu Ylönen. Subsequent development of the protocol suite proceeded in several developer groups, producing several variants of implementation. The protocol specification distinguishes two major versions, referred to as SSH-1 and SSH-2. The most commonly implemented software stack is OpenSSH, released in 1999 as open-source software by the OpenBSD developers. Implementations are distributed for all types of operating systems in common use, including embedded systems.
- PuTTY ('pʌti/) is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "PuTTY" has no official meaning.
- PuTTY was written and is maintained primarily by Simon Tatham, a British programmer.

https://en.wikipedia.org/wiki/Secure_Shell

<https://en.wikipedia.org/wiki/PuTTY>

Connect to adroit via SSH

- Open terminal (or using PuTTY)
- ssh username@adroit.princeton.edu

ssh-keygen

- ssh-keygen is a tool for creating new authentication key pairs for SSH. Such key pairs are used for automating logins, single sign-on, and for authenticating hosts.
- SSH supports several public key algorithms for authentication keys. These include:
 - rsa - an old algorithm based on the difficulty of factoring large numbers. A key size of at least 2048 bits is recommended for RSA; 4096 bits is better. RSA is getting old and significant advances are being made in factoring. Choosing a different algorithm may be advisable. It is quite possible the RSA algorithm will become practically breakable in the foreseeable future. All SSH clients support this algorithm.
 - dsa - an old US government Digital Signature Algorithm. It is based on the difficulty of computing discrete logarithms. A key size of 1024 would normally be used with it. DSA in its original form is no longer recommended.
 - ecdsa - a new Digital Signature Algorithm standarized by the US government, using elliptic curves. This is probably a good algorithm for current applications. Only three key sizes are supported: 256, 384, and 521 (sic!) bits. We would recommend always using it with 521 bits, since the keys are still small and probably more secure than the smaller keys (even though they should be safe as well). Most SSH clients now support this algorithm.
 - ed25519 - this is a new algorithm added in OpenSSH. Support for it in clients is not yet universal. Thus its use in general purpose applications may not yet be advisable.

ssh-keygen and GitHub

- On your machine,

```
$ ssh-keygen -t ecdsa -b 521
```

- don't skip to set *passphrase*

```
$ cat .ssh/id_ecdsa.pub
```

- copy the public key
- ssh to adroit

```
$ vi .ssh/authorized_keys
```

- type "i" and paste, ":wq"

```
$ ssh-copy-id <YourNetID>@<cluster-name>.princeton.edu
```

```
$ chmod 700 ~/.ssh
```

```
$ chmod 600 ~/.ssh/id_ecdsa
```

- Repeat `ssh-keygen` on adroit
- GitHub → Settings → SSH and GPG keys → New SSH key → paste

Git and Version Control

before we move on...

- create an account on GitHub - <https://github.com>
- cat .ssh/id_ecdsa.pub – copy
- add SSH-key: GitHub → Settings → SSH and GPG keys → New SSH key → paste your public key
- create following files and folders on the adroit machine

```
project/
  my_project/
    file1.py
  tests/
    test1.py
  README.md
```

before we move on...

- go to <https://github.com/pmelchior/usrp-sciprog>
- fork it to your account
- go to the forked repo
 - click code — click ssh — click copy button
 - git clone git@github.com:YOUR_GITHUB_USERNAME/usrp-sciprog.git

A screenshot of a GitHub repository page for `pmelchior/usrp-sciprog`. The page shows a list of commits from the `main` branch. A large blue arrow points from the top right towards the **About** section on the right side of the page.

Code | Issues (2) | Pull requests | Actions | Projects | Wiki | Security | Insights

Go to file | Add file | **Code**

`main` | 9 branches | 0 tags

Author	Commit Message	Date
pmelchior	Update LICENSE	4c3319f 15 days ago
	cleanup	15 days ago
	Update SSH_instructions.md	2 years ago
	Solutions added	2 years ago
	daily agenda, titles	3 years ago
	minor tweaks to optimization notebook	2 years ago
	cleanup of .DS_Store files	3 years ago
	Update LICENSE	15 days ago
	Fix typo in Tea's talk	10 months ago
	update to latest anaconda	15 days ago
	activities update	3 years ago
	Add data as a file (#95)	12 months ago
	Add data as a file (#95)	12 months ago

About

Material for the Princeton Undergraduate Summer Research Program in Astrophysics

Readme | MIT license | 15 stars | 14 watching | 96 forks

Releases

No releases published

Packages

No packages published

Contributors 44

