```
<CH.1>[Introduction to Linux]
-Linux is a Kernel:
  'A kernel is the core or central controller of the operating system.'
  'The kernel is loaded at boot time and stays running to manage'
    'every aspect of the functioning system.'
  'GNU/Linux defines the operating system.'
  'GNU is free software that provides open source equivalents of many'
    'common UNIX commands and building tools.'
  "GNU's Not UNIX"
-Linux is Open Source
-Linux has Distributions:
  'The kernel, tools, and suite of applications that come bundled together.'
  'Usually include a package manager (dpkg, RPM,  ... )'
  'Many distros suited to different tasks (running servers,'
    'desktops, statistical computing, ... )'
```

```
<CH.2>[Operating Systems]
  'Software that runs on a computing device and manages the hardware'
    'and software components that make up a functional system.'
  'Schedule programs to run in a multi-tasking manner.'
  'Provide standard services to users and programs.'
-Decision Points:
  'Role; Function; Stability; Compatibility; Cost'
  'Life Cycle (Release Cycle, Maintenance Cycle)'
  'Interface (CLI vs GUI)'
    'CLI: Command Line Interface'   (command-based)
    'GUI: Graphical User Interface' (event-based)
-Microsoft Windows:
  'Not UNIX based.'
-macOS:
  'UNIX with some proprietary code.'
-Linux:
  'UNIX-like'
  'A distro:'
    >'Sets up the storage'
    >'Builds the kernel'
    >'Install hardware drivers, applications, and utilities'
-Linux Distros:
  >'Red Hat' (Enterprise Linux)
    -'fedora' (for devs)
    -'CentOS Stream' (preview release of RHEL)
    -'Alma Linux/Rocky Linux' (replacement for CentOS)
  >'SUSE'    (Enterprise Linux)
  >'Debian'  (Community)
    -'ubuntu' (for everybody)
  >'Android'
    -'Uses Linux and ART (android runtime)'
    -'ART introduces AOT (ahead-of-time) compilation.'
  >'Raspbian' (for the Raspberry Pi)
-Embedded Systems:
  'Linux emphasis on small size and power consumption'
  "DVR's, smart TV'S, etc.."
  "IoT potential (networks, sensors, actuators) and AI integration."
```

```
<CH.3>[Working in Linux]
-'Get familiar with Linux as a desktop environment'
-Getting to the command line:
  'The command line interface (CLI) is a simple text input system'
  ' for entering anything from single-word commands to complicated scritps.'
    >'Ordinary command line task are starting programs, parsing scripts,'
      'and editing text files for system or admin config.'
  'GUI terminal or virtual terminal'
-Applications:
  'kernel: airtraffic controller'  |  'applications: planes'
  'The kernel decides  which programs get which blocks of memory, it starts and'
    'kills applications, and handles displaying text or graphics on a monitor.'
  'Applications make requests to the kernel and in turn receive resources:'
    'memory, CPU, disk space.'
  "Applications follow the kernel's API:"
    "Behave as if they have a single block of memory in the system."
  'The kernel remaps memory blocks, shares memory with applications,'
    'and can even swap to disk.'
  'The kernel handles application switching (scheduling or multi-tasking.)'
  'A process is a task that is loaded and tracked by the kernel.'
  'An application may need multiple processes to function properly.'
-Major Applications:
  "server: primarily runs data on other's behalf."  |  "desktop: direct user interaction"
  "Linux can simulate almost all aspects of a production environment,"
    "from development to testing, to verification on scaled-down hardware."
  >server:
    -"no direct interaction with the monitor and keyboard of the machine it runs."
    -"serve information to other computers called clients."
  >desktop:
    -"web browsers, text editors, music players... "
  >tools:
    -"software to manage computer systems easier."
    -"compilers, configure displays, provide a shell to Linux users... "
<Server Applications:>
  'Linux excels at running server applications because of its reliability and efficiency.'
-Web Server:
  'hosts content for web pages.'
  'viewed by a web browser using the HyperText Transfer Protocol (HTTP)'
    'or its encrypted version (HTTPS.)'
  'e.g.: Wordpress, Apache (using the Apache HTTPD daemon), NGINX, etc.'
-Private Cloud Servers:
  'provide private cloud software meeting needs of security, privacy, and regulatory compliance.'
    >'ownCloud: GNU AGPLv3 licence'  | 'NextCloud: GNU AGPLv3'
-Database Servers:
  'stores information and allows for easy retrieval and querying.'
  'e.g.: MariaDB, MySQL, Firebird, PostgreSQL.'
  'some use a language called Structured Query Language (SQL.)'
-Email Servers:
  'tasks required for email:'
    >'Mail Transfer Agent (MTA): software used to transfer electronic messages'
      'to other systems (Postfix >> Sendmail.)'
      >>SMTP (Simple Mail Transfer Protocol) server
    >'Mail Delivery Agent (MDA): stores the email in the users mailbox'
      '(usually invoked from final MTA.)'
      >>POP(Post Office Protocol)/IMAP (Internet Message Access Protocol) server
        'Dovecot, Cyrus IMAP'
  'Microsoft Exchange ships as a software package/suite with all required components.'
  'in the closed source world individuals have very few options to make.'
  'in the open source world many options can be modularly included.'
-FIle Sharing:
  'Samba: allows a Linux machine to look and behave like a Windows one, sharing files'
    'and participating in a windows domain.'
  'Netatalk: Linux performs as an Apple Macintosh file server.'
  'UNIX/Linux uses the Network File System (NFS), which is part of the kernel.'
  'the Domain Name System (DNS) converts a url string to an ip address.'
  'the Lightweight Directory Access Protocol(LDAP) can store information such as'
    'user accounts and security roles easily searchable.'
  'openLDAP is the main used program in Linux.'
  'Dynamic Host Configuration Protocol (DHCP) listens for requests and assign a'
    'free address from the DHCP pool.'
<Dektop Applications>
-Email:
  'Thunderbird: full-featured desktop email client (connects to IMAP/POP,'
    'displays emails locally, and sends emails through SMTP).'
  'Evolution: GNOME'  | 'KMail: KDE'
  'switching between email clients without losing data is possible thanks to'
    'IMAP/POP standardization and local email formats.'
-Creative:
  'Blender: 3D movie creation.'
  'GIMP (GNU Image Manipulation Program): 2D image manipulation.'
  'Audacity: audio editing.'
-Productivity:
  'OpenOffice: Apache foundation open-source application suite.'
  'LibreOffice: OpenOffice fork. It can be integrated with wiki software providing a'
    'powerful intranet solution.'
  'LibreOffice Calc: akin to MS Excel.'
  'LibreOffice Writer: akin to MS Word. Ability to link to Calc spreadsheets and'
    'display updated information.'
  'LibreOffice Impress: akin to MS PowerPoint.'
-Web Browsers:
  'Mozilla Firefox'  |  'Google Chrome'
  'Firefox: open source, fast, feature-rich, excellent support for web developers.'
  'configuring settings options can limit the amount of info browsers share while searching the web.'
<Console Tools>
  'considerable overlap between the skills of software development and system administration.'
-Shells:
  'accepts shell commands, like file manipulations and starting applications, and'
    'to pass those to the Linux kernel for execution.'
  'Bourne shell: Bourne Again Shell(Bash): users who are comfortable with it can'
    'operate effectively on most Linux System.'
  'C shell: tcsh'
  'Korn shell (ksh)'
  'Z shell (zsh)'
-Text Editors:
  'vi (or vim)' | 'emacs' : Powerful tools to edit text files
  'pico '       | 'nano'  : Provide very basic text editing
  'learn vi or vim before is too late'
<Package Management>
  'every Linux system needs to add, remove, and update software.'
  'takes care of which files belong to which package and even dowloads updates from repositories.'
-Debian Package Management:
  'dpkg'
  'Aplication Package Tool (apt-get) [front-end program to the dpkg tool]'
  'other front-ends include: aptitute and GUI (Synaptic, Software Center.)'
-RPM Package Management:
  'the Linux Standards Base (a Linux Foundation project) specifies standards to'
    'increase compatibility between Linux systems.'
  'RPM is the standard package management system (according to LSB.)'
  'yum, up2date (old front-end tools.)'
  'dnf (newer front-end tool.)'
  'back-end program: interacts with other programs.'
  'front-end program: interacts with people.'
  'GUI front-end: Yumex (dead), dnfdragora (newer), Gnome Package Kit.'
  'there also exists the ZYpp method (zipper command as the basis.)'
  'If a command affects the state of a package, administrative access is required'
    '(root privileges required for most package management commands.)'
  'a query or a search can be perfomed by a regular user, but it cant add or remove any packages.'
<Development Languages>
  'computer programming languages provide a way to enter instructions in'
    'a more human readable format.'
    1-'interpreted: translates written code into computer code as the program runs.'
    2-'compiled: translates all at once before execution.'
  >'C: Linux is written in C. It maps closely to the generated machine code.'
  >'C++: object-oriented C (abstraction programming paradigm: `objects` that contain'
    '`data` and `code`.)'
  >'Objective C: oop language that added Smalltalk-style messaging to the C language.'
  >'Java: oop language that compiles to the Java Virtual Machine (JVM.) running on any computer.'
  >'JavaScript: high-level, interpreted, and one of the core technologies of the world wide web.'
  >'Perl: interpreted, built for text manipulation, now used for automation and web applications.'
  >'PHP: create dynamic web pages (usually with Apache.)'
    '[websites: WordPress(blogging), cacti(monitoring.)]'
  >'Ruby: influenced by Perl. [automation tools: Chef, Puppet.]'
  >'Python: scripting language with excellent statistical processing abilities, a favorite'
    'in academia [Django Web app framework as well.]'
  'a programming library bundles common tasks into a distinct package that can'
    'be used by the developer.'
    >'ImageMagick: manipulate image in code.'
    >'OpenSSL: cryptographic library'
    >'C library: provides low-level set of basic instructions, such as reading and'
      'writing to files and displays.'
<Security>
  'cookies: small piece of data stored on the users computer.'
    'It is the primary mechanism websites use to track users.'
  'limit the cookies and tracking pixels you accept (blocking them entirely or'
    'clearing them out regularly.)'
-Password Issues:
  'password management apps: programs that store login credentials in encrypted form.'
  'two-factor authentication (2FA): a password is supplemented by a second "factor",'
    'usually a code sent to the users phone.'
-Protect yourself:
  'browsing the web leaves a digital footprint.'
  'good passwd: at least 10 characters long, numbers, letters, and special characters.'
  'KeePassXC: open source, encrypted, and cross platform password manager.'
  'check for updates regularly.'
  'protect against accepting incoming connections.'
  'Gufw: graphical interface to Ubuntus Uncomplicated Firewall (UFW)[using iptables.]'
  'Firewalld: Daemon and graphical interface for configuring network and firewall zones and rules.'
-Privacy tools:
  'Encryption: e.g.: HyperText Transfer Protocol Secure (HTTPS.)'
  'Virtual Private Network: create an encrypted channel of communication between two systems.'
  'the Tor browser works by relaying requests through a network of servers that'
    'prevents websites knowing the users identity.'
-The Cloud:
  'cloud: computing resources from one or many off-site data centers which'
    'can be accessed over the internet.'
    >'delegate management of IT infrastructure to a third-party.'
  'cloud deployment models:'
    1>Public cloud: 'a provider offers cloud services to the general public'
      '(Amazon, Google, etc.)'
    2>Private cloud: 'cloud infrastructure set up for the sole use of a particular organization'
      '(Rackspace or IBM.)'
    3>Community cloud: 'set up and used by a group of organizations with'
      'common goals or requirements.'
    4>Hybrid cloud: 'contains multiple clouds (public, community, or private.)'
-Linux in the Cloud:
```

```
'powers 90% of the public cloud workload.'
  1>Flexibility:  'provide IT resources quickly and at any time. Linux is modular'
    'by design and highly adaptable.'
  2>Accessibility: 'accessed from anywhere over a network by any device'
    '(desktop, mobile, thin client.)'
  3>Cost-Effective: 'eliminates overhead from underutilized resources.'
  4>Manageability: 'automated management tools: frees up administrators time.'
  5>Security: 'Linux is one of the most secure and reliable operating systems.'
    'The community upholds the robust reputation of Linux.'
  6>Virtualization:
    'the process in where one physical computer (host),'
      'runs multiple copies of an operating system (guests.)'
    'Linux is a multi-user operating system.'
    'guest images can be pre-configured for specific functions to allow rapid deployment.'
    'not necessary to run same operating system on all hosts.'
  7>Containers and Bare Metal Deployment:
    'Docker and Kubernetes: containerization technologies allowing apps to'
      'run in a serverless environment (eg: database processing, storage, etc.)'
    'containers are organized in `pods` that run within a `node`'
      '(allowing node-to-node communication) controlled by the `master` node.'
    'ability to create the systems of the future.'
```

```
<CH.4>[Open Source Software and Licensing]
  'source code: human readable set of computer instructions.'
  'closed source license: right to use the machine code, but no seeing or'
    'tampering with the source code.'
  'open source software: right to obtain the software source code, and to'
    'expand and modify programs for their own use.'
  'standards organizations: IEEE (Institute of Electrical and Electronics Engineers)'
    'and POSIX (Portable Operating Systems Interface.)'
-Open Source Licensing:
  1>Ownership: 'who owns the ip behind the software.'
  2>Money Transfer: 'how money changes hands.'
  3>Licensing: 'what do you get? what can be done with the software?'
  'End User License Agreement (EULA): closed-source, only binary copies distributed'
    'and usually for one device only.'
  'GNU General Public License version 2 (GPLv2.):'
    'open-source, distribute it, get it back. This is the Linux license.'
-The Free Software Foundation:
  'goal of promoting free(libre) software, freedom to share, study, and modify the'
    'underlying source code.'
  'copyleft: a method for making a program or other work `libre` and requiring all'
    'modified and extended versions of the program free as well.'
  'GPLv3 prevents tivoization (running free software in closed hardware.)'
  'Lesser versions allow developers to use and integrate open-source components in'
    'their projects without having to release proprietary source code.'
-The Open Source Initiative:
  'Open Source Initiative (OSI): believes software should be freely available but no'
    'restrictions should be imposed on its use.'
  'dont provide any licenses but classify them.'
  'BSD (Berkeley Software Distribution):'
    'provide 2 licenses, no copyleft (neither does MIT)'
    '[permissive free software license.]'
  'Free and Open Source Software (FOSS) or FLOSS (Free/Libre/Open.)'
```

```
-Creative Commons:
  'address the intentions behind FOSS licenses for non-software entities.'
    1>Attribution   (BY): 'gives author credit without implying his endorsement.'
    2>ShareAlike    (SA): 'allows others to copy, distribute, perform, and'
      'modify work on the same terms.'
    3>NonCommercial (NC): 'same as share alike excluding commercial activities.'
    4>NoDerivatives (ND): 'copy, distribute, display, and perform.'
      'Modify only with creators permission.'
  'CC0: creative commons version of public domain.'
-Open Source Business Models:
  'GPL doesnt prohibit selling software.'
```

```
<CH.5>[Command Line Skills]
  'Most consumer OSes are designed to shield the user from the ins and outs of the CLI.'
  'The Linux community positively celebrates the CLI for its power and speed.'
<Shell>
  'The shell is the command line interpreter that translates commands entered by a user'
    'into actions performed by the operating system.'
  'Bash shell popular features:'
    >Scripting: 'place commands in a file and then interpret them all commands.'
    >Aliases: 'create short nicknames for longer commands.'
    >Variables: 'store information for the shell and user.'
  'the prompt indicates that commands can be run and'
    'also provides useful information to the user.'
  sysadmin@localhost:~$
    'sysadmin: user name',
    'localhost: system name',
    '~: home dir:/home/sysadmin'
<Commands>
  'a software program that, when executed on the CLI, performs an action on the computer.'
  'typical command format:'
    command [options] [argument]
-Arguments:
  'can be used to specify something for the command to act upon.'
  'the ls command accepts multiple arguments.'
-Options:
  'can be used with commands to expand or modify the way a command behaves.'
  'using the -l option in the ls command results in a long listing.'
  'can be combined such as in -lh (long-listing human-readable.)'
  'order of combined options isnt important'
  'single word: -h'
  'full-word: --human-readable'
```

```
-History:
  'commands are stored in a history list after execution.'
  history   →'view history list.'
  !n        →'execute nth command in history list.'
  history n →'outputs n previous commands.'
  !-n       →'execute nth from the bottom of history list.'
<Variables>
  'are given names and stored temporarily in memory.'
  'allow the user or the shell to store data.'
-Local:
  'Local (shell) variables exists only in the current shell and cannot affect'
    'other commands or applications.'
  'lowercase by convention and associated with with user-based tasks.'
  'input: variable=value'
  'output: echo $variable (display output in the terminal)'
-Environment:
  'or global variables, are available system-wide (all shells and performing tasks.)'
  'PATH, HOME, and HISTSIZE are some the global variables.'
  'env: outputs a list of the environment variables.'
    'ej: env | grep variable1'    'the pipe | character passes the output of'
                                  'the env command to the grep command.'
  'export: turns a local variable into an environment variable.'
  'unset: removes exported variables.'
-Path:
  'PATH: contains a list that defines which directories the shell looks in'
    'to find commands.'
<Command Types>
  'type: can be used to determine information about command type.'
-Internal:
  'also called built-in command, are built into the shell itself.'
    $type cd
    'cd is a shell builtin'
```

```
-External:
  'stored in files searched by the shell.'
  which command 'displays the full path to the command.'
    $which cal
    'cal is /usr/bin/cal'
  type -a command 'displays all locations of the command.'
-Aliases:
  alias name=command 'creates a new alias for the command.'
  'aliases created this way only persist while the shell is open.'
-Functions:
  'can be built using existing commands'
-Quoting:
  'information within quotes are treated in a specific way.'
  'Glob characters, also called wildcards, are symbols that have special'
    'meaning to the shell.'
  1>Double(""):
    'doesnt convert the glob patterns:' (*, ?, [])
    'still allow for command substitution:' $(command) or `command`
    'or variable substitution:' $[variable]
  2>Single(''):
    'prevent the shell from doing any interpreting of special characters.'
  3>Backquotes(``):
    'or backticks, are used to specify a command within a'
      'command (command substitution.)'
    $echo today is `date`
    'today is Wed Mar 24 01:55:15 PM AST 2021'
<Control Statements>
  'allow the use of multiple commands at once.'
  1>Semicolon:
    command1; command2; command3
    'used to run a command right after the other, regardless if'
      'the past command failed.'
```

```
2>Double Ampersand:
  command1 && command2
  'acts as a short-circuited logical `AND`'
3>Double Pipe:
  command1 || command2
  'acts as a short-circuited logical `OR`'
```

```
<CH.6>[Getting Help]
<Man Pages>
  'manual pages (name from UNIX) are used to describe the features of a command.'
-Viewing Man Pages:
  man [command]
  'The man command uses a pager to display documents: `less` & `more`.'
  'use' H 'key while viewing a man page to display help.'
-Section Within Man Pages:
  'man pages are broken into sections:'
    >NAME:          'name and brief description'
    >SYNOPSIS:      'example(s) of how the command is executed'
    >DESCRIPTION:   'more detailed description'
    >OPTIONS:       'lists options as well as a description of how they are used'
    >FILES:         'lists the files associated and a description on using them'
    >AUTHOR:        'man page creator'
    >REPORTING BUGS:'how to report bugs'
    >COPYRIGHT:     'basic copyright information'
    >SEE ALSO:      'provides an idea of where you can find additional info'
-Searching Man Pages:
  'type'  / 'followed by a search term, then hit ENTER.'
  'press' n 'to move to the next match.'
  'press' Shift+n 'to move to the previous match.'
-Man Pages Categorized by Sections:
  'there are thousands of man pages on a typical Linux distro.'
  'to organize all of these man pages, they are organized by section:'
    1-General Commands
    2-System Calls
    3-Library Calls
    4-Special Files
    5-File Formats and Conventions
    6-Games
    7-Miscellaneous
    8-System Administration Commands
    9-Kernel Routines
  man -f [COMMAND] 'search for man pages by name.'
  man -k [COMMAND] 'search for man pages by both name and description.'
  whatis  'is the same as' man -f 'on most distros.'
  apropos 'is the same as' man -k 'on most distros.'
<Finding Commands and Documentation>
  whatis 'or' man -f 'command returns what section a man page is stored in.'
-Where Are These Commands located:
  whereis 'searches for the location of a command or its man pages.'
  'man pages are typically compressed with `gzip` ending in `.gz`'
-Find Any File or Directory:
  locate command 'finds any file or directory.'
    -'searches a database of all files and directories that were on the system'
      'when the database was created.'
    -locate -c 'returns the count of files matching.'
    -locate -b 'returns files matching the "basename".'
    -locate -b '"\search_term"`: returns files mathing exactly their "basename".'
  updatedb 'updates the "locate" database (root access required.)'
<Info Documentation>
  info command 'also provides documentation in a slightly different format than `man`.'
    -'it gives a logical organizational structure, making it easier to read.'
-Viewing Info Documentation:
  'current node is displayed highlighted in the bottom.'
  u ->'go up one node.'
  n ->'go to the next node.'
  l ->'go back to the last node seen in this window'
  q ->'quit info altogether'
-Navigating Info Documents:
  'hiting' SHIFT+H 'while reading the info documentation displays a'
    'listing of movement commands.'
-Exploring Info Documentation:
  'execute info command without any arguments. good starting point to explore its features'
```

```
<Additional Sources of Help>
-Using the Help Option:
  command --help 'provides basic information, very similar to the synopsis in man pages'
-Aditional System Documentation:
  '`readme` files: provide additional documentation.'
  '`/usr/share/doc` or `/usr/doc`'
    'is where sys admins go to learn how to set up more complex software services.'
```

```
<Ch.7> Navigating the Filesystem
  'Everything is considered a file in Linux.'
  'Files store data and programs.'
  'Directories are a type of file used to store other files.'
<Directory Structure>
  'also called a filesystem.'
  'the top level directory is called `root(/)`.'
  'no drives in Linux; each physical device is accessible under a directory,'
    'as opposed to a drive letter.'
  -Home Directory:
    '/home/username is the path to the home directory'
    'is one of the few directories where the user has full control to create'
      'and delete additional files and directories.'
    'represented by the ~ (tilde, virgulilla) character.'
  -Current Directory:
    pwd [OPTIONS] 'prints the working directory (current location within the filesystem)'
  -Changing Directories:
    cd [OPTIONS] [PATH] 'changes directories'
    'cd without path takes you to the users home directory'
<Paths>
  'a path is a list of directories separated by the / character.'
  'paths are the directory addresses and can be used to indicate the location of'
    'any file within the filesystem.'
  -Absolute Paths:
    'allow the user to specify the exact location of a directory.'
    'always begin at the root directory (/).'
  -Relative Paths:
    'start with a name of a directory contained within the current directory.'
  -Shortcuts:
    ..->'parent directory'
    . ->'current directory'
```

```
<Listing Files>
  ls [OPTIONS]... [FILE]...
  'when used with no options or arguments, it lists the files in the current directory.'
  'place a backlash (\) character to escape an alias (avoid using.)'
-Hidden Files:
  ls -a [FILE]
  'usually configuration or customization files.'
-Long Listing:
  ls -l [FILE]  'shows files metadata in a details view.'
   >'file type:'
     d ->'directory'
     - ->'regular file'
     l ->'symbolic link'
     s ->'socket'
     p ->'pipe'
     b ->'block file'
     c ->'character file'
   >'permissions:'
     r ->'read'
     w ->'write'
     x ->'execute'
   >'hard link count:' ->'how many hard links point to this file?'
   >'user owner'
   >'group owner'
   >'file size:'
     -'for directories, it describes the amount of reserved bytes to keep track of'
      'filenames in the directory (ignore it.)'
   >'timestamp'
   >'filename'
-Human-Readable Sizes:
  ls -lh [FILE]
  'adding' -h 'to the' ls 'command presents the file size in a more'
    'human-readable format (like MB or GB.)'
  'the -h option must be used with the -l option'
```

```
-List Directories:
  ls -d
  'refers to the current directory (. directory) but not the contents within it.'
-Recursive Listing:
  ls -R [FILE]
  'recursive listing: diplay all of the files in a directory as well as'
    'all the files in all subdirectories under that directory.'
-Sort a Listing:
  ls -S [FILE]        'sorts files by size'
  ls -t [FILE]        'sorts files by timestamp (modification date)'
  ls -t --full-time   'displays complete timestamp (assumes -l option by default)'
  ls -r               'performs a reverse sort'
```

```
<CH.8>[Managing Files and Directories]
<Globbing>
  'wild cards'
  'allow you to specify patterns that match filenames in a directory.'
  'interpreted by the shell before running any command.'
-Asterik '*':
  'used to represent ⩾0 characters in a filename.'
-Question Mark '?':
  'represents any single character. It matches exactly one character.'
-Brackets '[]':
  'match a single character by representing a range of characters that are'
    'possible match characters.'
  [am]*  →'matches any file that begins with `a`' or `m`
  [p-t]* →'matches all files between `p`' and '`t` (inclusive.)'
  [0-9]* →'matches any file that contain at least a number.'
  '`[9-0]` doesnt work because it doesnt follow the ASCII table range.'
-Exclamation Point '!':
  'used in conjunction with the square brackets to negate a range.'
  [!DP]* →'matches any file that does NOT begin with a D or P.'
-Listing with Globs:
  'the shell expands the glob pattern into corresponding file names.'
  'when using the' ls 'command, always use the option' -d 'to display the'
    'name of directories instead of their content.'
<Copying files>
  cp [source] [destination]  'no output when successful (no news is good news!)'
-Verbose mode:
  -v →'causes the cp command to produce output if successful.'
-Avoid overwriting data:
  -i →'requires you to answer' y 'or' n 'for every copy that could end up'
    'overwriting an existing files content.'
  -n →'no clobber, no overwrite'
-Copying Directories:
  cp    →'does not copy directories by default.'
  cp -r →'copies both files and directories (recursively.)'
```

```
<Moving Files>
  mv [source] [destination]
  'users need permissions to remove files from a directory.'
-Renaming files:
  mv 'is also used to rename files.'
  'if a destination directory is not specified, the file is renamed using the'
    'destination file name and remains in the source directory.'
-Additional move options:
  -i`  →'interactive'
  -n`  →'no clobber'
  -v`  →'verbose'
  'there is no' -r 'as it does it by default.'
<Creating Files>
  touch [OPTION] FILE   →'creates a file with 0 bytes (no data.)'
  touch me              →'creates a file called me xD'
<Removing Files>
  rm [OPTION] [FILE]    →'remove files'
  'the files are permanently deleted. No command to undelete a file.'
  'use the' -i 'option when caution is required.'
-Removing Directories:
  rm -r [FILE] →'remove directories and their contents recursively.'
  rmdir        →'only deletes empty directories.'
<Creating Direcories>
  mkdir [OPTION] DIRECTORY  →'make (create) directory'
```

```
<CH.9>[Archiving and Compression]
  'Archiving:   combine multiple files into one.'
  'Compressing: Makes the files smaller by removing redundant information.'
<Compressing Files>
  'Compression reduces the amount of data needed to store or transmit a file.'
  'a compression algorithm is a procedure to encode the original file,'
    'and as a result, make it smaller.'
    1-'Lossless:  No information is removed from the file'
    2-'Lossy:     Information might be removed from the file.'
  gzip [OPTION]... [FILE]... →'compress files'
    ⇒'Lempel-Ziv compression algorithm'
  gzip file →'compresses file to file.gz (deletes original file)'
  gzip -l   →'list relevant fields:'
             '(compressed, uncompressed, ratio, uncompressed_name)'
  gunzip or gzip -d   →'expand files (decompress)'

  'other commands:'
  bzip2  & bunzip2  a →'block-sorting file compressor'
    ⇒'Burrows-Wheeler algorithm (can compress files smaller than gzip at'
      'the expense of more CPU time.)'
  xz  & unxz  →'compress or decompress .xz or .lzma files.'
    ⇒'Lempel-Ziv-Markov (LZMA) chain algorithm, results in lower'
      'decompression CPU times on par with gzip and better compression than bzip2'
```

```
<Archiving Files>
  tar →'TApe aRchive'
  'the' tar 'command takes in several files and creates a single output file that'
    'can be split up again into the original files.'
  tar 'modes:'
    1-Create  'make a new archive out of a series of files.'
    2-Extract 'pull one or more files out of an archive.'
    3-List    'show contents without extracting.'
-Create Mode:
  tar -c [-f ARCHIVE] [OPTIONS] [FILE... ]
  tar -czf  'create and compress (using gzip)  [.tar.gz | .tgz]'
  tar -cjf  'create and compress (using bzip2) [.tar.bz2 | .tbz | .tbz2]'
  tar -cJf  'create and compress (using xz) [.tar.xz | .txz]'
-List Mode:
  tar -t [-f ARCHIVE] [OPTIONS]
  tar -tjf 'list files in an archive compressed with bzip2'
-Extract Mode:
  tar -x [-f ARCHIVE] [OPTIONS]
  tar -xjvf 'extract verbosely with bzip2'
<ZIP Files>
  zip [OPTIONS] [zipfile [file... ]]
  zip file.zip file →'compress file into file.zip'
  unzip -l          →'lists files in the .zip archive'
  unzip file.zip    →'decompresses `file.zip`'
```