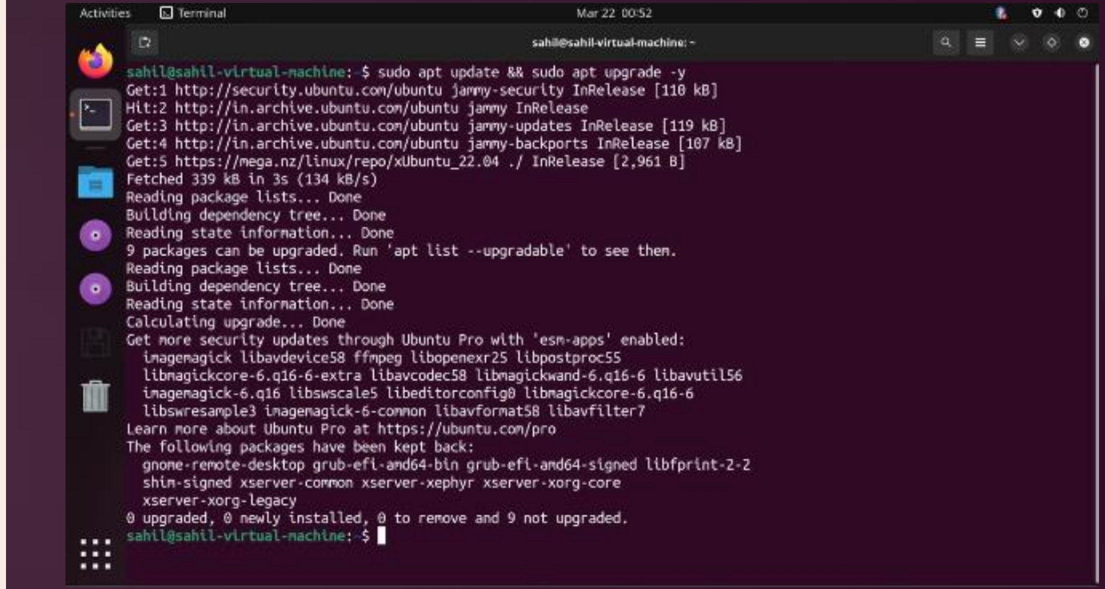
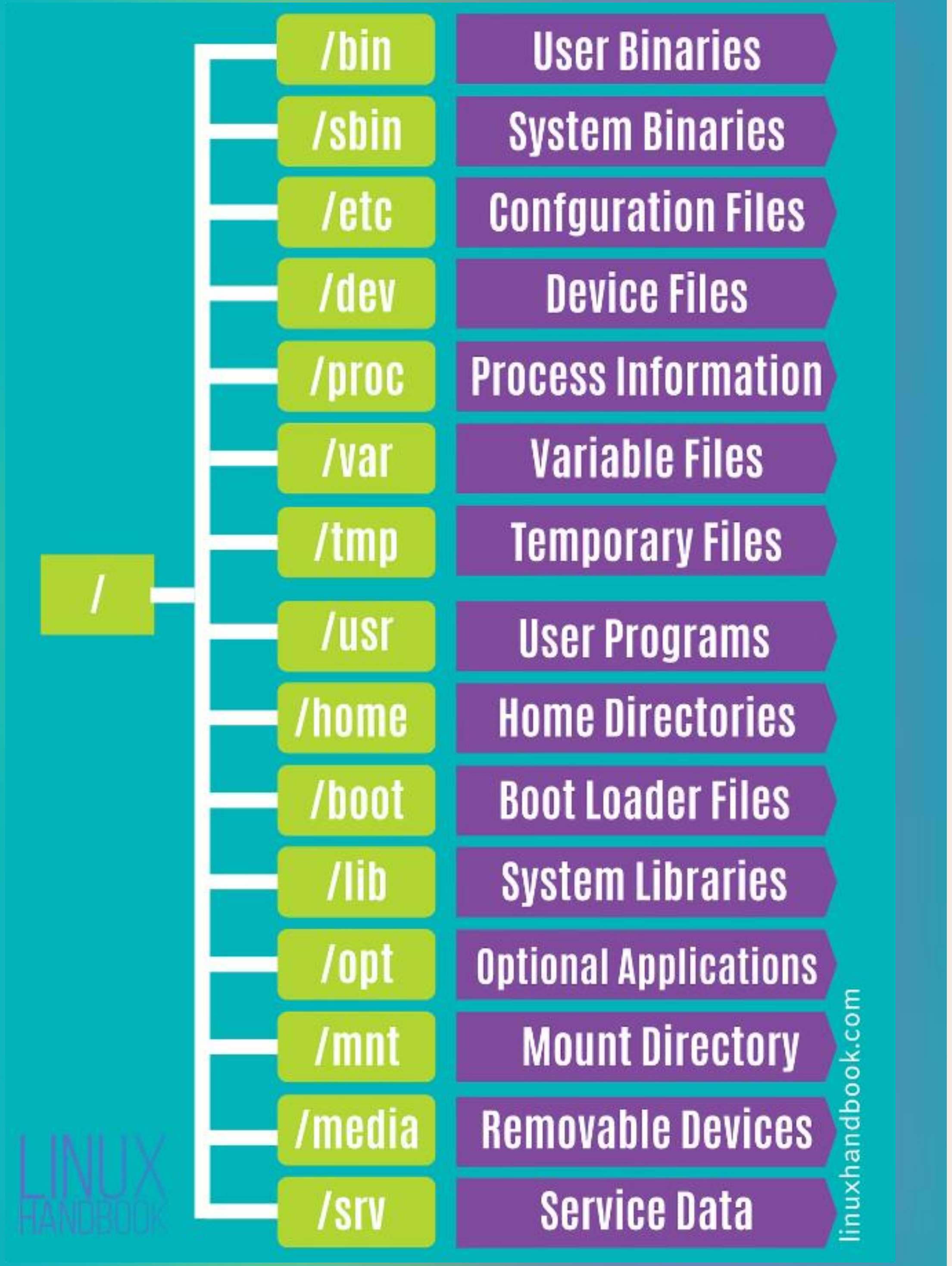
Globbing Options in Linux

Globbing, short for "global substitution," is a powerful feature in Linux shells that allows you to match files and directories using wildcard characters. It simplifies tasks like selecting multiple files for operations like copying, deleting, or moving.

by Pratham Borghare



```
sahil@sahil-virtual-machine:~$ sudo apt update && sudo apt upgrade -y
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:5 https://mega.nz/linux/repo/xUbuntu_22.04 ./ InRelease [2,961 B]
Fetched 339 kB in 3s (134 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  imagemagick libavdevice58 ffmpeg libopenexr25 libpostproc55
  libmagickcore-6.q16-6-extra libavcodec58 libmagickwand-6.q16-6 libavutil56
  imagemagick-6.q16 libswscale5 libeditorconfig0 libmagickcore-6.q16-6
  libswresample3 imagemagick-6-common libavformat58 libavfilter7
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following packages have been kept back:
  gnome-remote-desktop grub-efi-amd64-bin grub-efi-amd64-signed libfprint-2-2
  shim-signed xserver-common xserver-xephyr xserver-xorg-core
  xserver-xorg-legacy
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
sahil@sahil-virtual-machine:~$ 
```



linuxhandbook.com

What is Globbing?

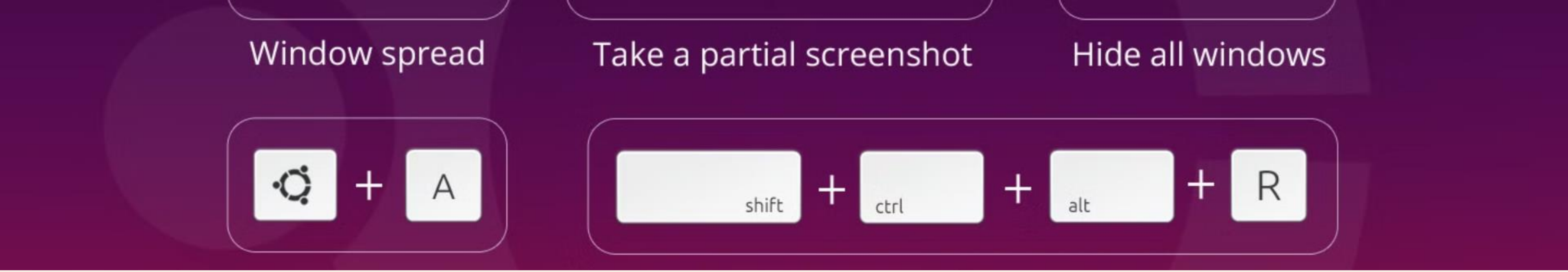
Globbing is a pattern matching mechanism that simplifies file selection in Linux shell commands. It uses special characters, known as wildcards, to match file names and directories based on their patterns.

File Selection

Globbing enables the selection of multiple files with similar names, saving time and effort.

Shell Commands

Globbing integrates seamlessly with shell commands, making it a versatile tool for file management.



Globbing Special Characters

Globbing uses special characters known as wildcards, which represent any character or sequence of characters within a filename.

1

Asterisk (*)

Matches any sequence of characters, including zero characters.

2

Question Mark (?)

Matches any single character.

3

Bracket ([ ])

Matches any single character within the specified range.

4

Brace ({ })

Expands to multiple file names based on the provided patterns.

Globbing with Wildcards

Wildcards are the cornerstone of globbing, providing flexibility in pattern matching.

Asterisk (*)

The asterisk (*) matches zero or more characters.

Example: `*.txt` matches all files ending with ".txt".

Question Mark (?)

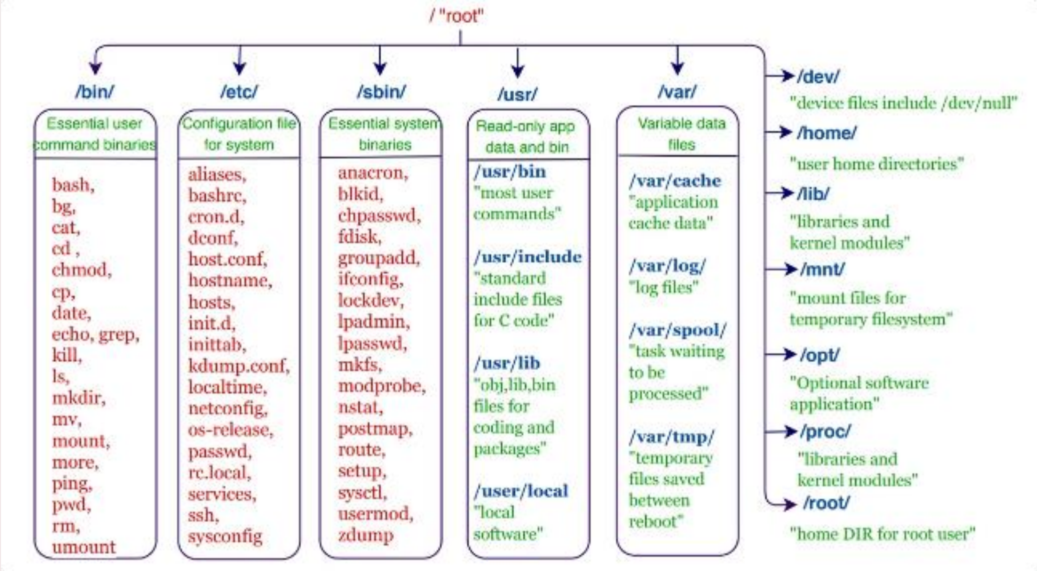
The question mark (?) matches any single character.

Example: `report?` matches `report1`, `report2`, `reportA`, etc.

Globbing with Character Ranges

Character ranges allow you to specify a set of characters to match within a filename.

Character Range	Description	Example
[a-z]	Matches any lowercase letter from a to z.	files in `[a-z]*.txt`
[A-Z]	Matches any uppercase letter from A to Z.	files in `[A-Z]*.jpg`
[0-9]	Matches any digit from 0 to 9.	files in `report[0-9]*.pdf`


/ "root"
/bin/
Essential user command binaries
bash, bg, cat, cd , chmod, cp, date, echo, grep, kill, ls, mkdir, mv, mount, more, ping, pwd, rm, umount
/etc/
Configuration file for system
aliases, bashrc, cron.d, dconf, host.conf, hostname, hosts, init.d, inittab, kdump.conf, localtime, netconfig, os-release, passwd, rc.local, services, ssh, sysconfig
/sbin/
Essential system binaries
anacron, blkid, chpasswd, fdisk, groupadd, ifconfig, lockdev, lpadmin, lpasswd, mkfs, modprobe, nstat, postmap, route, setup, sysctl, usermod, zdump
/usr/
Read-only app data and bin
/usr/bin "most user commands"
/usr/include "standard include files for C code"
/usr/lib "obj,lib,bin files for coding and packages"
/user/local "local software"
/var/
Variable data files
/var/cache "application cache data"
/var/log/ "log files"
/var/spool/ "task waiting to be processed"
/var/tmp/ "temporary files saved between reboot"
/dev/ "device files include /dev/null"
/home/ "user home directories"
/lib/ "libraries and kernel modules"
/mnt/ "mount files for temporary filesystem"
/opt/ "Optional software application"
/proc/ "libraries and kernel modules"
/root/ "home DIR for root user"

```
root@linux:~$
```

Globbing with Negation

Negation allows you to exclude specific characters or ranges from the pattern.

1

Negation with `!`

The exclamation mark (!) negates the following character range.

2

Example

`[!0-9]*.txt` matches files ending with ".txt" that don't have a digit at the start of their name.

Globbing with Alternatives

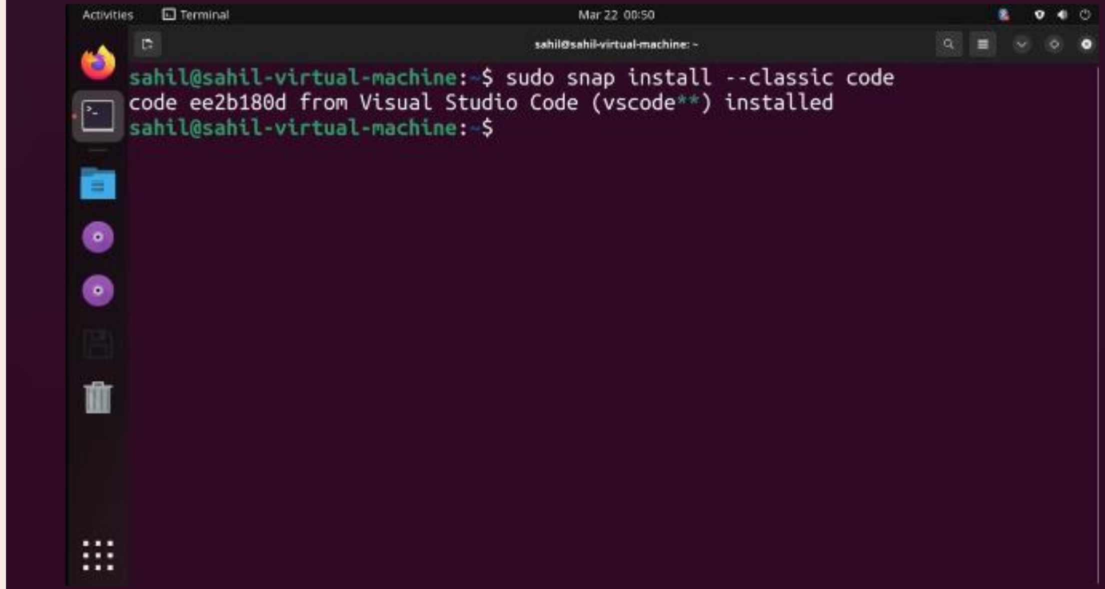
Alternatives allow you to specify multiple patterns to match.

1 Braces ({ })

Braces are used to create multiple alternative patterns within a single globbing expression.

2 Example

`file{1,2,3}.txt` matches `file1.txt`, `file2.txt`, and `file3.txt`.



```
sahil@sahil-virtual-machine:~$ sudo snap install --classic code
code ee2b180d from Visual Studio Code (vscode**) installed
sahil@sahil-virtual-machine:~$
```

Globbing with Recursive Patterns

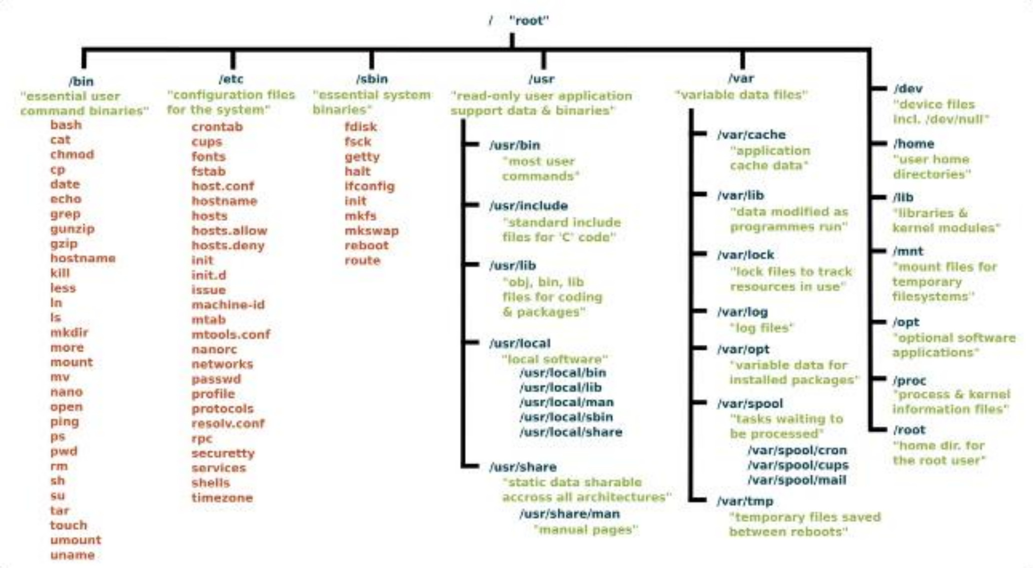
Recursive patterns allow you to match files within subdirectories.

Double Asterisk (**)

The double asterisk (**) matches any number of directories.

Example

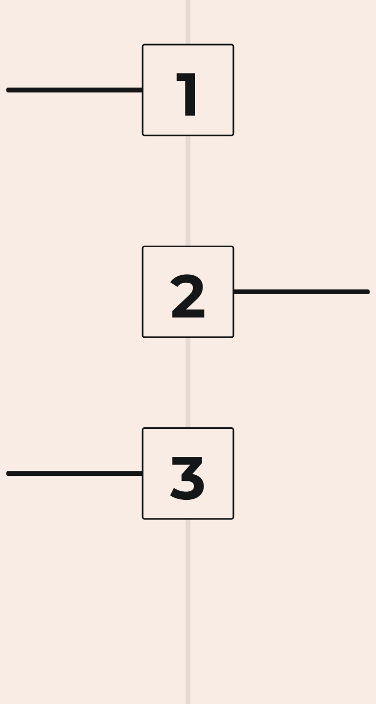
`**/*.txt` matches all ".txt" files in the current directory and its subdirectories.



```
➜  ~ /usr/local/bin/python3 /Users/flavio/dev/python/program.py
Flavio
➜  ~
```

Globbing with Extended Patterns

Extended globbing features provide more advanced pattern matching capabilities.



1. **Bash Option**

Enabling extended globbing in bash shell with `shopt -s extglob`.

2. **Pattern Matching**

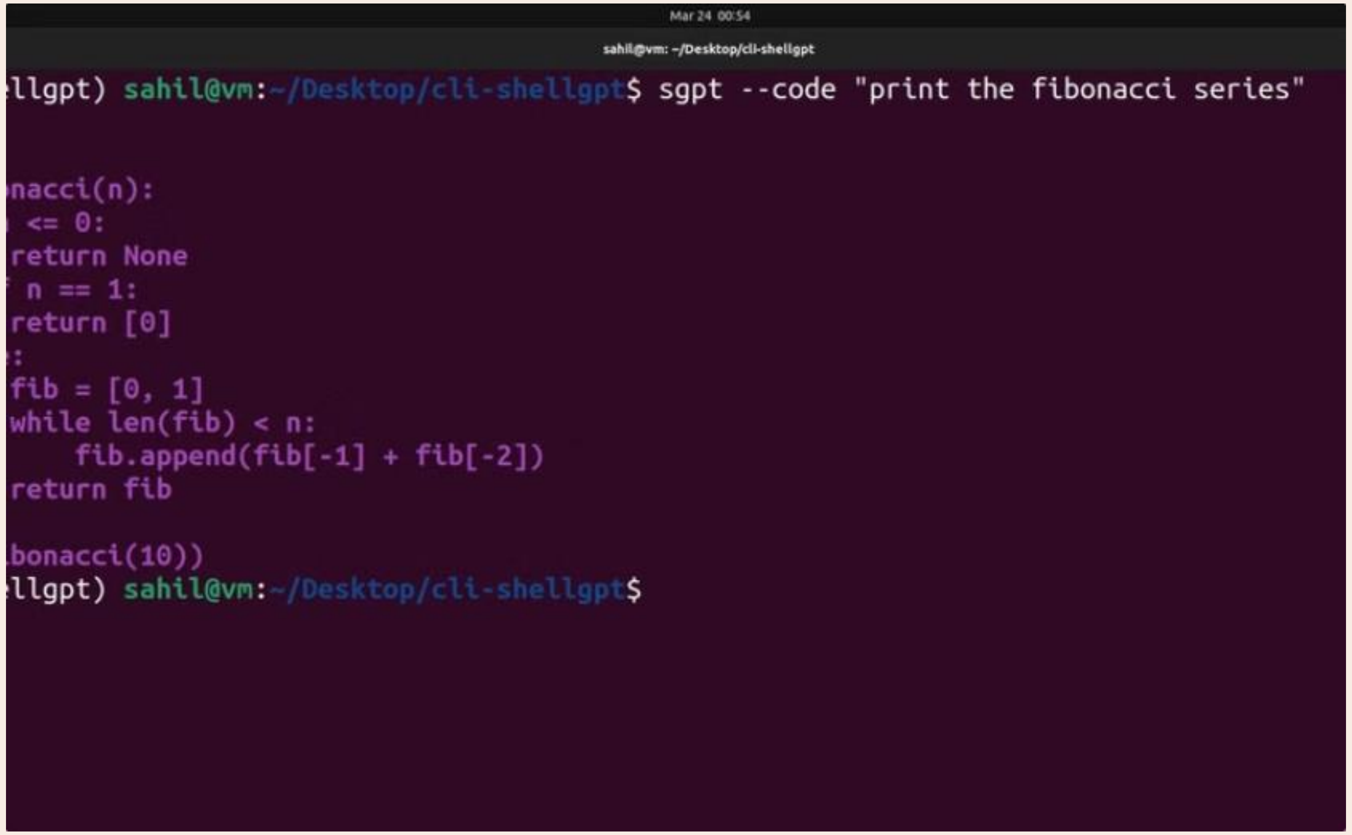
Extended patterns allow for more complex matching criteria beyond basic wildcards.

3. **Examples**

Matching files with a specific prefix: `*(pattern)`.

Globbing Best Practices

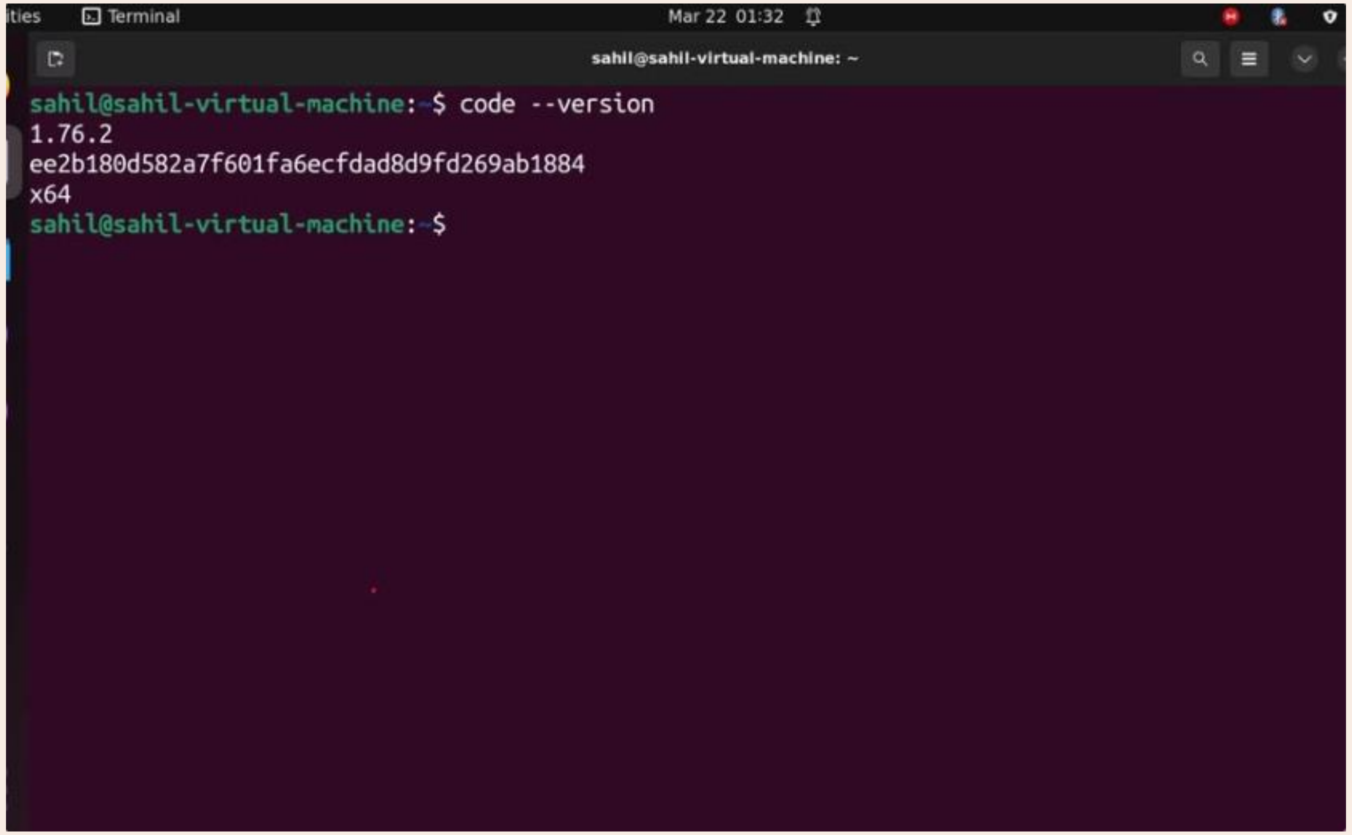
Following best practices ensures efficient and safe globbing usage.



```
llgpt) sahil@vm:~/Desktop/cli-shellgpt$ sgpt --code "print the fibonacci series"


nacci(n):
 <= 0:
 return None
 n == 1:
 return [0]
:
 fib = [0, 1]
 while len(fib) < n:
     fib.append(fib[-1] + fib[-2])
 return fib

bonacci(10))
llgpt) sahil@vm:~/Desktop/cli-shellgpt$
```



```
sahil@sahil-virtual-machine:~$ code --version
1.76.2
ee2b180d582a7f601fa6ecfdad8d9fd269ab1884
x64
sahil@sahil-virtual-machine:~$
```

Be Specific

Use precise globbing patterns to avoid accidental selection of unwanted files.

Double-Check

Verify the output of globbing expressions before executing potentially destructive commands.