

BASH

Unix Philosophy

According to Doug McIlroy

- Make each program do one thing well.
- So, to do a new job, build afresh rather than complicate old programs by adding new features (otherwise known as “bells and whistles”).
- Machine shop vs. appliance (gives you the tools and you to make appliance)



Top 50 Linux Commands you must know

- | | | | | |
|-----------|------------|-------------|----------------------|---------------------------|
| 1. ls | 11. cat | 21. diff | 31. kill and killall | 41. apt, pacman, yum, rpm |
| 2. pwd | 12. echo | 22. cmp | 32. df | 42. sudo |
| 3. cd | 13. less | 23. comm | 33. mount | 43. cal |
| 4. mkdir | 14. man | 24. sort | 34. chmod | 44. alias |
| 5. mv | 15. uname | 25. export | 35. chown | 45. dd |
| 6. cp | 16. whoami | 26. zip | 36. ifconfig | 46. whereis |
| 7. rm | 17. tar | 27. unzip | 37. traceroute | 47. whatis |
| 8. touch | 18. grep | 28. ssh | 38. wget | 48. top |
| 9. ln | 19. head | 29. service | 39. ufw | 49. useradd |
| 10. clear | 20. tail | 20. ps | 40. iptables | 50. passwd |

What is Bash?



- ✓ Bash is the default shell, or command language interpreter, in the GNU operating system.
- ✓ Bash = acronym for the (Steve) 'Bourne-Again SHell',
- ✓ a conformant implementation of the IEEE POSIX Shell and Tools specification (IEEE Working Group 1003.2).
- ✓ Portable - it currently runs on nearly every version of Unix.

Pipes in Linux

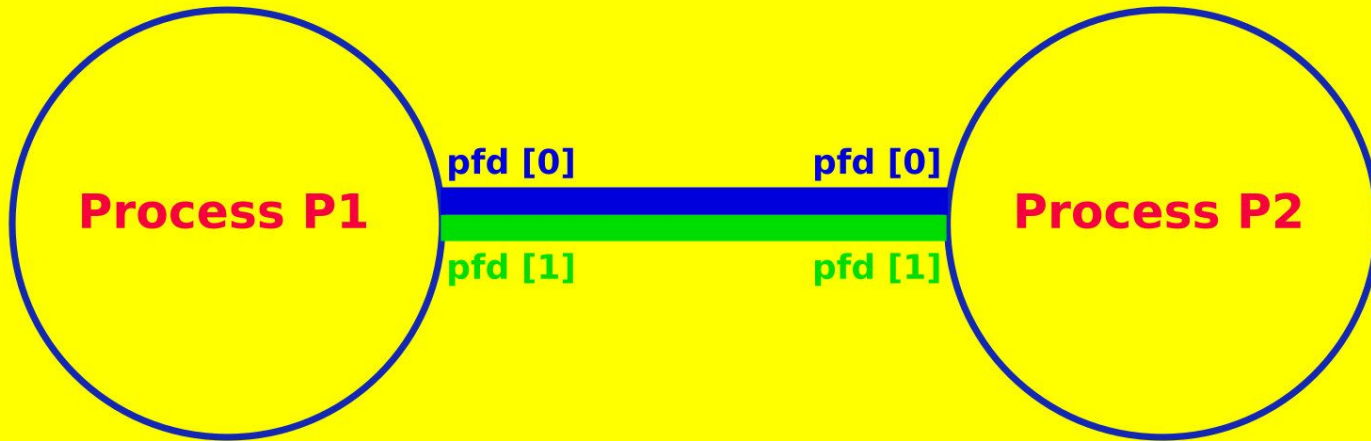
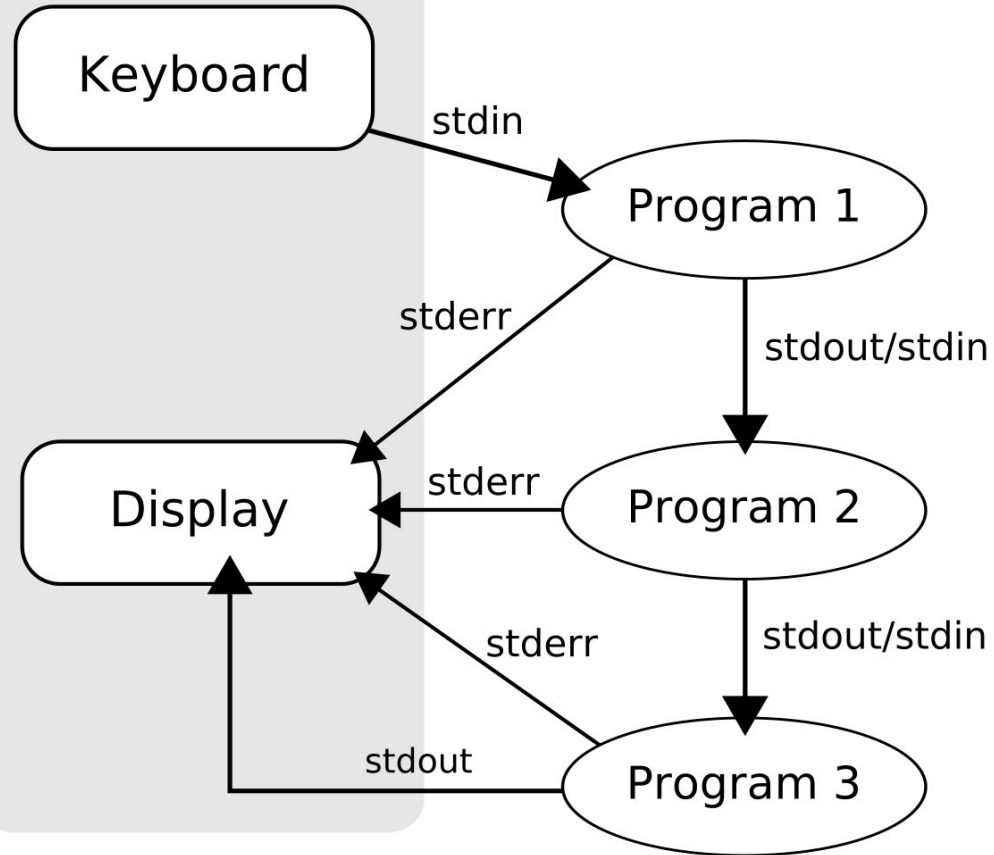
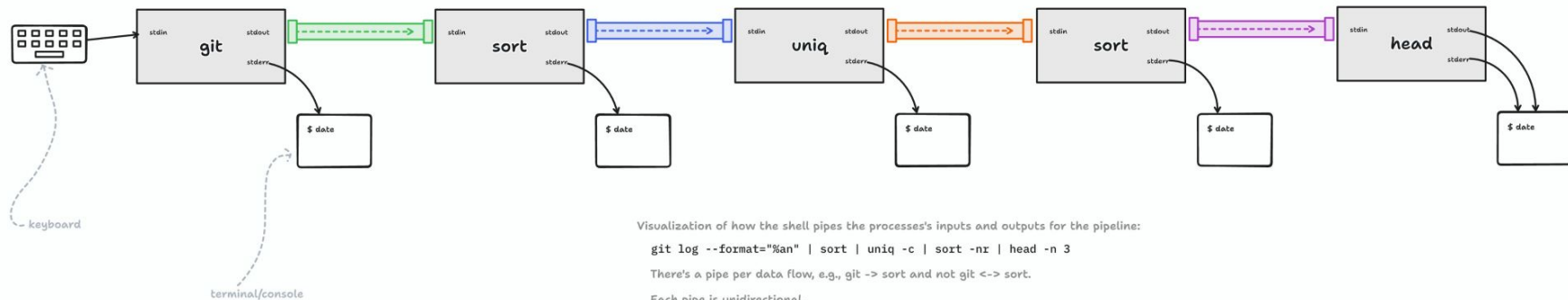


Fig 1: Two processes connected with a pipe

Text terminal



Visualization of pipelining in Unix



```
rishabh@rishabh: ~/GFG
rishabh@rishabh:~/GFG$ ls -l | more
total 28
drwxrwxr-x 2 rishabh rishabh 4096 Jan 29 21:11 demo1
-rw-rw-r-- 1 rishabh rishabh  26 Jan 25 23:03 demo1.txt
drwxrwxr-x 2 rishabh rishabh 4096 Jan 29 21:11 demo2
-rw-rw-r-- 1 rishabh rishabh   0 Jan 25 23:04 demo2.txt
drwxrwxr-x 2 rishabh rishabh 4096 Jan 29 21:11 demo3
-rw-rw-r-- 1 rishabh rishabh   0 Jan 25 23:04 demo.txt
-rw-rw-r-- 1 rishabh rishabh 123 Jan 26 16:02 sample1.txt
-rw-rw-r-- 1 rishabh rishabh  44 Jan 26 15:52 sample2.txt
-rw-rw-r-- 1 rishabh rishabh   0 Jan 26 00:12 sample3.txt
-rw-rw-r-- 1 rishabh rishabh  26 Jan 25 23:03 sample.txt
rishabh@rishabh:~/GFG$
```


Using wget

bash

Copy

```
curl -s https://www.nasa.gov/ \  
  | grep -Eo '<img [^>]*src="[^"]+"' \  
  | sed 's/.*src="\([^"]*\)".*/\1/' \  
  | grep -Ei '\.(jpg|png)$' \  
  | xargs -n 1 wget -q -nc
```

Explanation:

```
curl -s https://www.nasa.gov/ \  
  | grep -Eo '<img [^>]*src="[^"]+"' \  
  | sed 's/.*src="\([^"]*\)".*/\1/' \  
  | grep -Ei '\.(jpg|png)$' \  
  | xargs -n 1 wget -q -nc
```

```
# Download images (only jpg and png)
```

```
curl -s https://www.nasa.gov/ \  
  | grep -Eo '<img [^>]*src="[^"]+"' \  
  | sed 's/.*src="\([^"]*\)".*/\1/' \  
  | grep -Ei '\.(jpg|png)$' \  
  | xargs -n 1 wget -q -nc
```

```
# Generate thumbnails for downloaded images (assuming they're in the current directory)
```

```
mkdir -p thumbnails
```

```
for img in *.jpg *.png; do
```

```
  [ -f "$img" ] && convert "$img" -thumbnail 150x150 "thumbnails/$img"
```

```
done
```

```
curl -s
https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_b
y_revenue | pup 'table tr json{' | jq -r '.[
| [ (.children[0].text // ""),
  (if .children[1].children then .children[1].children[0].text else .children[1].text
end // ""),
  (.children[2].text // ""),
  (.children[3].text // ""),
  (.children[4].text // ""),
  (.children[5].text // "") ] | @csv'
```

4 Advantages of Bash

A central rounded rectangle contains the text '4 Advantages of Bash'. Four arrows point outwards from the corners of this rectangle to four separate text blocks arranged in a square pattern around it. The background is a solid yellow color.

Access your
operating system
more efficiently

Input and
output with
text files

Automate
with ease

Run any
programming
language from the
same application

Function definition

```
hello_world() {  
    echo "Hello, World!"  
}
```

Variables

```
greeting="Hello"  
target="World"
```

Print variables using parameter expansion

```
echo "$greeting, $target!"
```

Conditional statement

```
if [[ $greeting == "Hello" ]]; then  
    echo "The greeting variable is correctly set to 'Hello'."  
else  
    echo "The greeting variable is not set to 'Hello'."  
fi
```

Array declaration

```
fruits=("Apple" "Banana" "Cherry")  
echo "Here is a list of fruits:"
```

Looping through an array

```
for fruit in "${fruits[@]"; do  
    echo " - $fruit"  
done
```

Copy

```
# Calling the function defined earlier
echo "Now calling our function:"
hello_world

# Reading user input
read -p "What is your name? " user_name
echo "Nice to meet you, $user_name!"

# Using a simple arithmetic operation
num1=5
num2=3
sum=$((num1 + num2))
echo "Did you know? $num1 + $num2 = $sum"

# End of the script
echo "Script execution complete."
```