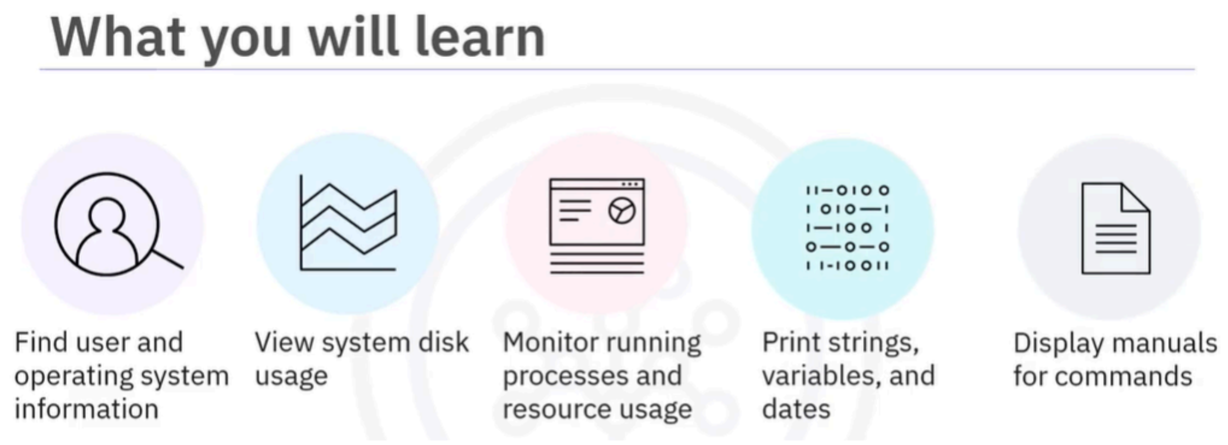
Informational Command



What you will learn

Find user and operating system information

View system disk usage

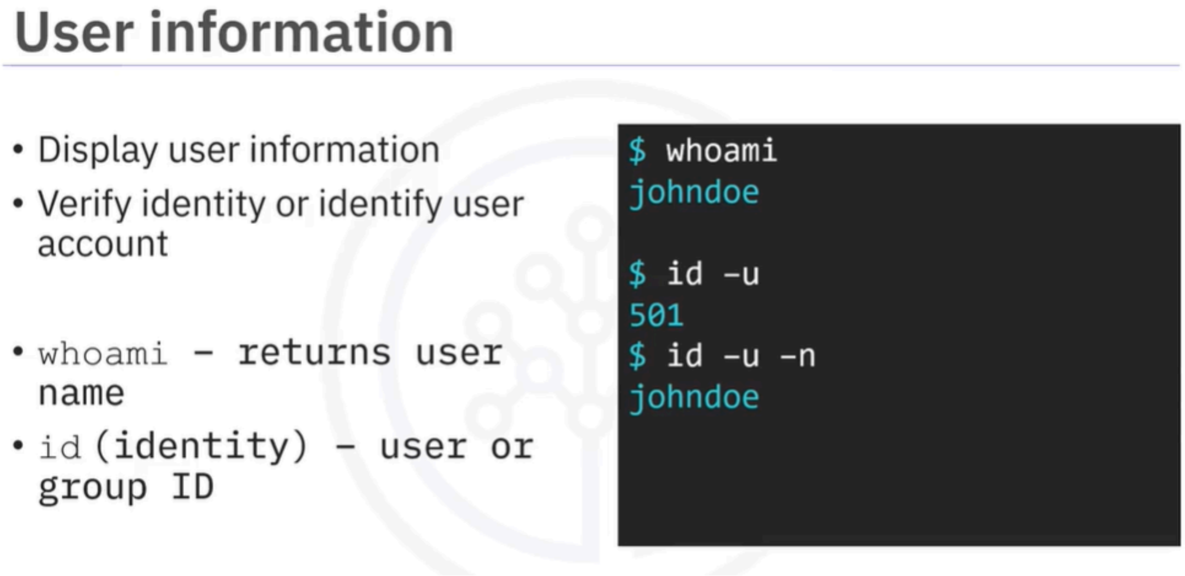
Monitor running processes and resource usage

Print strings, variables, and dates

Display manuals for commands

Informational commands are like tools in a toolbox that help you gather important information about your computer system. For example, if you want to know who is currently using the computer, you can use the command `whoami`, which simply tells you the username of the person logged in. Think of it as asking your friend, "Hey, who are you?" and they reply with their name.

Another useful command is `df`, which stands for "disk free." This command shows you how much space is available on your computer's hard drive. Imagine your hard drive as a bookshelf, and `df` helps you see how many books (files) you have, how many are on the shelf, and how much space is left for new books. By using these commands, you can easily monitor your system and keep it running smoothly!



User information

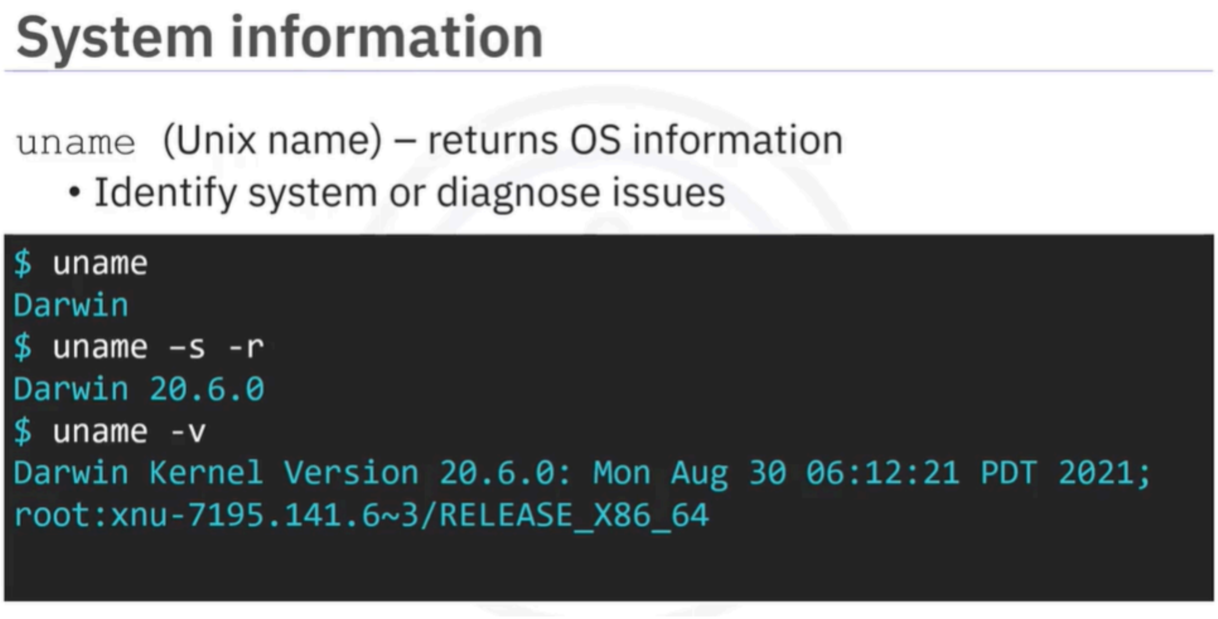
- Display user information
- Verify identity or identify user account

- whoami – returns user name
- id (identity) – user or group ID

```
$ whoami
johndoe

$ id -u
501
$ id -u -n
johndoe
```

- You can use informational commands in your terminal to display user information about your system. User info commands are useful when you need to verify the identity of the current user or determine which user account is running a particular command or process. For example, the command “whoami” displays the current user’s username. The “whoami” command takes no arguments and has no options. Here you see a usage example for someone logged in as user “John Doe.” Similarly, the “id” command returns the user or group ID, which is a number assigned to each user or group in the Linux system. In this example, using “id” with the “minus u” option returns the numerical ID of the user. If you would like the name corresponding to the numerical user ID, add the “minus n” option.



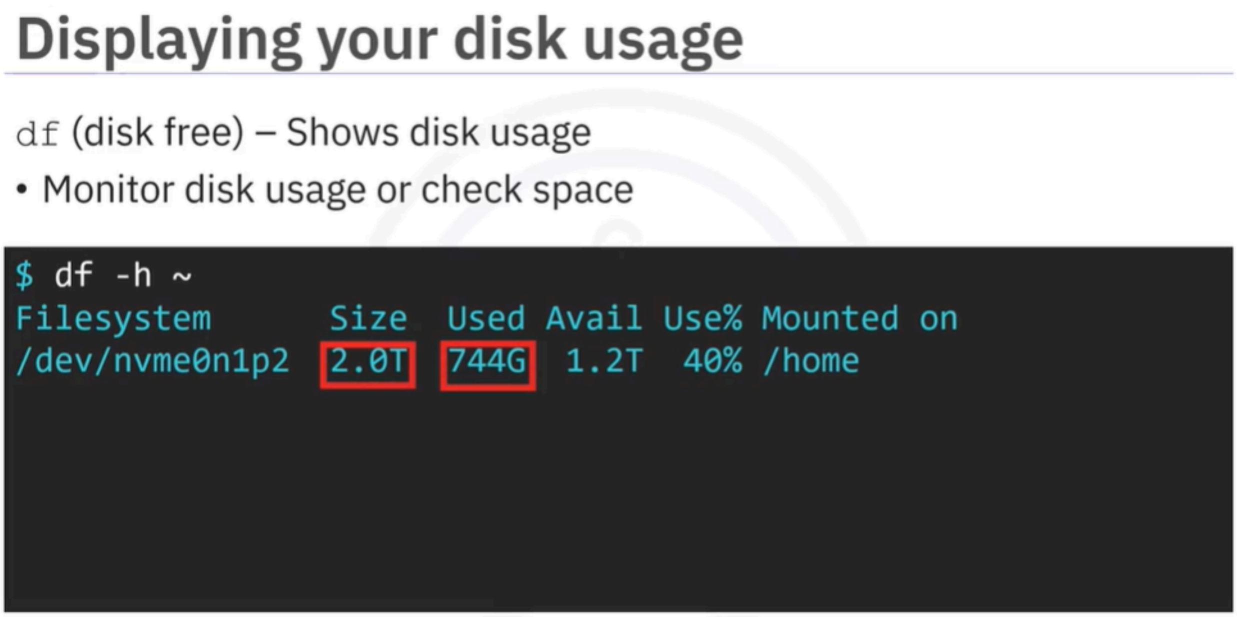
System information

`uname` (Unix name) – returns OS information

- Identify system or diagnose issues

```
$ uname
Darwin
$ uname –s -r
Darwin 20.6.0
$ uname -v
Darwin Kernel Version 20.6.0: Mon Aug 30 06:12:21 PDT 2021;
root:xnu-7195.141.6~3/RELEASE_X86_64
```

- The “uname” command, which stands for “Unix name”, returns OS information such as the kernel name and version number. This can be used to identify the type of system you are working on or diagnose system-related issues. Entering the uname command returns the name of the operating system, which in this case is Darwin. Including the “minus s” and “minus r” options returns both the OS name and its version. You can also view more detailed version information using the “minus v” option.



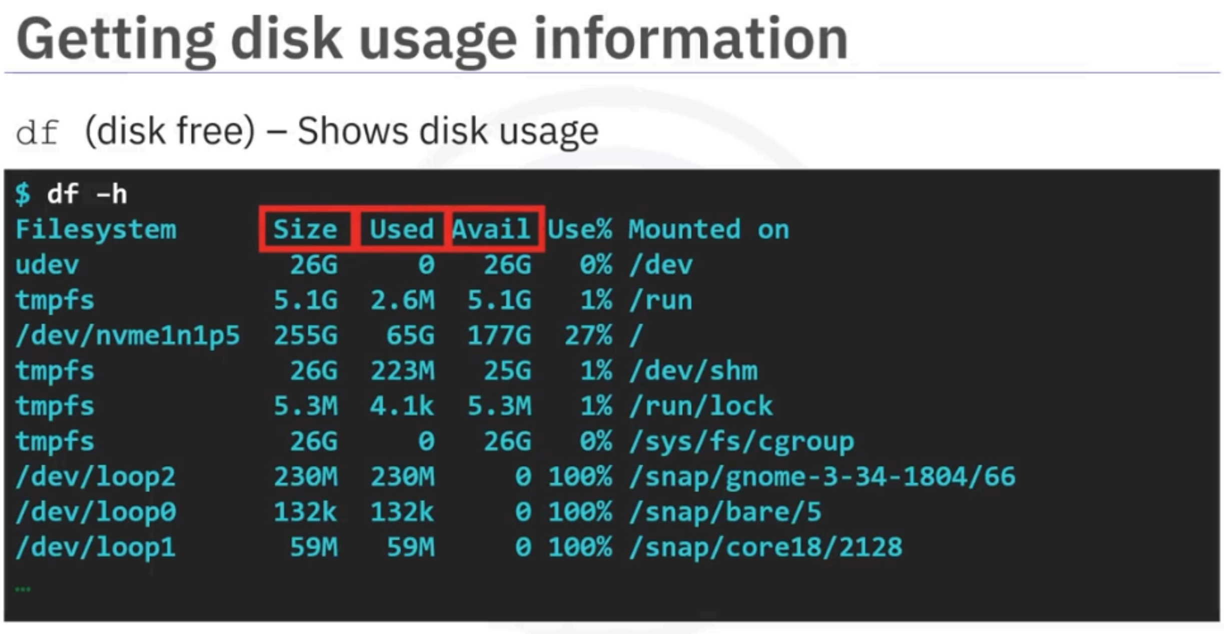
Displaying your disk usage

`df` (disk free) – Shows disk usage

- Monitor disk usage or check space

```
$ df -h ~
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p2  2.0T  744G  1.2T  40% /home
```

- You can use the “df”, or “disk free” command to display your system’s disk usage. It can be useful in situations where you need to monitor disk usage or check the available space on a particular file system. For example, entering “df minus h tilde”, displays the following table, which is specific to your home directory, represented by the tilde symbol. In this table, you can see all the disks mounted on your home directory. In Linux, you can “mount” a disk onto a directory, which means that the file system of that disk becomes accessible through that directory. The table also displays the percentage of storage used on each disk. The “minus h” option makes the output “human readable, expressing disk space in units like Gigabytes and Terabytes instead of Bytes.

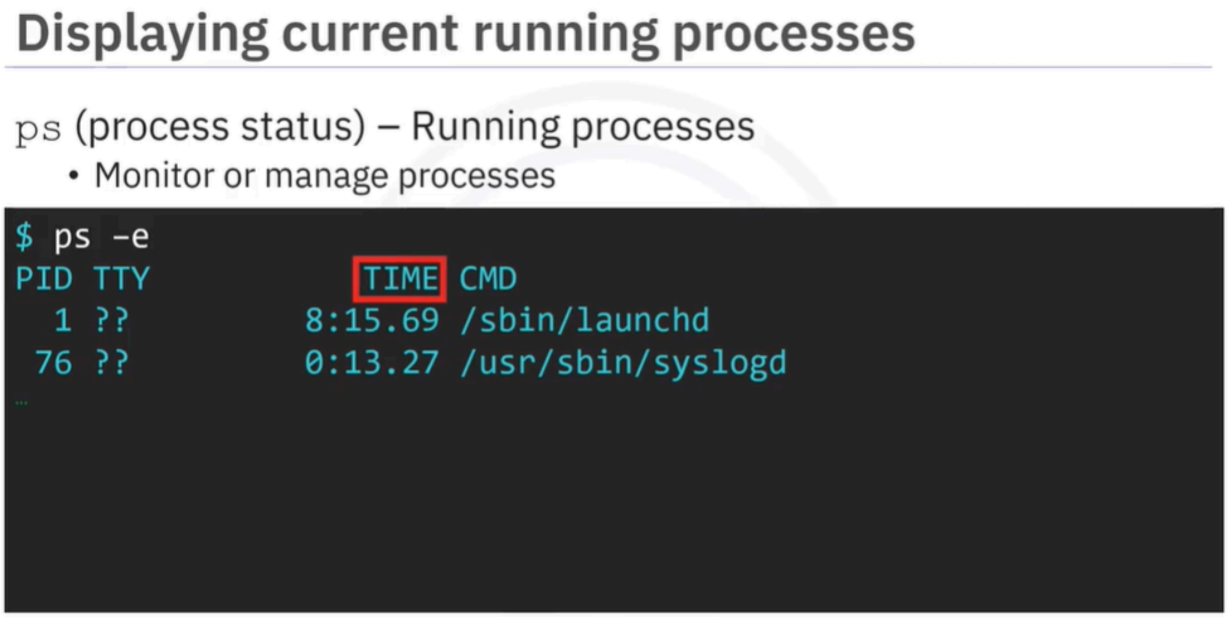


Getting disk usage information

df (disk free) – Shows disk usage

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev             26G     0   26G   0% /dev
tmpfs           5.1G  2.6M  5.1G   1% /run
/dev/nvme1n1p5  255G   65G  177G  27% /
tmpfs            26G  223M   25G   1% /dev/shm
tmpfs           5.3M  4.1k  5.3M   1% /run/lock
tmpfs            26G     0   26G   0% /sys/fs/cgroup
/dev/loop2      230M  230M     0 100% /snap/gnome-3-34-1804/66
/dev/loop0      132k  132k     0 100% /snap/bare/5
/dev/loop1       59M   59M     0 100% /snap/core18/2128
…
```

- To view disk usage on all filesystems, you can simply type “df minus h” without specifying a directory. The output includes the size, capacity used, and available space for each filesystem.



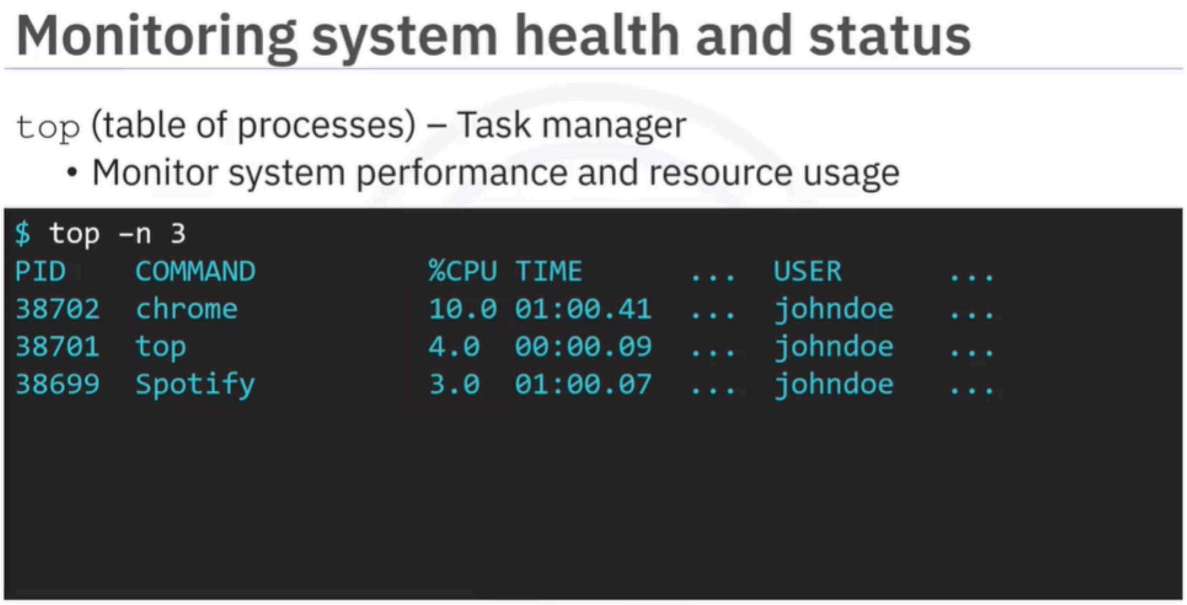
Displaying current running processes

ps (process status) – Running processes

- Monitor or manage processes

```
$ ps -e
PID TTY           TIME CMD
  1 ??         8:15.69 /sbin/launchd
 76 ??         0:13.27 /usr/sbin/syslogd
…
```

- To see the currently running processes on your system, you can use the “ps”, or “process status” command. This is helpful when you need to monitor or manage processes. Using “ps” with the “minus e” option will list all processes running on the system, regardless of which user started them. The “ps” command displays information such as the name of each running process, the process ID, and the time in minutes and seconds that each process has been running.



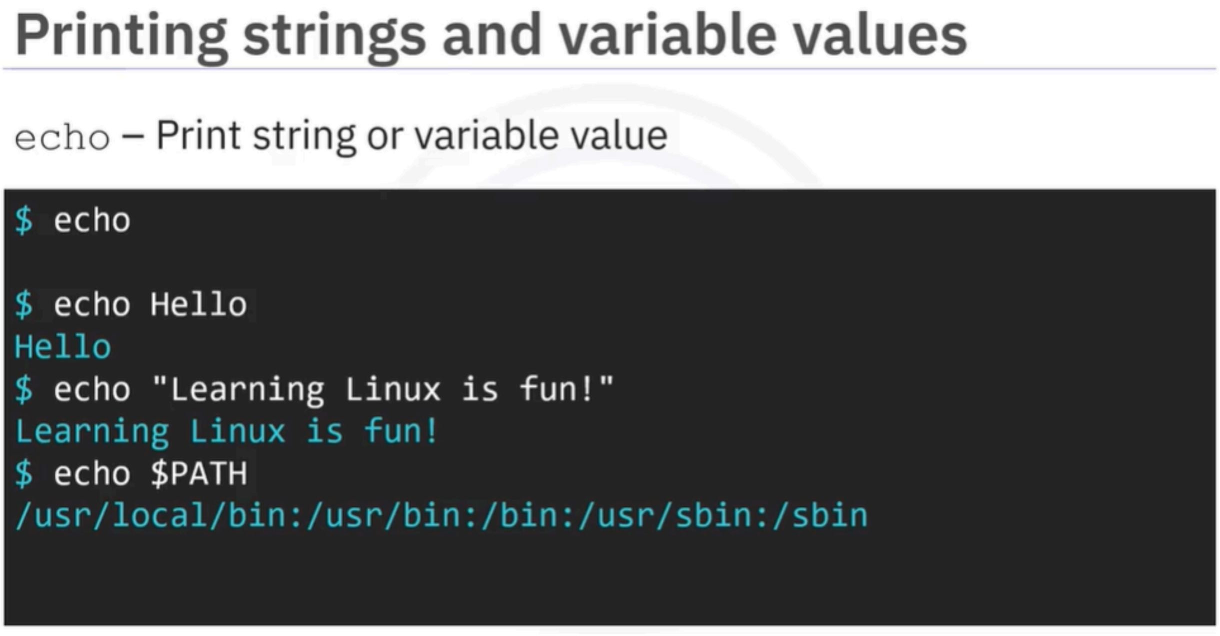
Monitoring system health and status

`top` (table of processes) – Task manager

- Monitor system performance and resource usage

```
$ top -n 3
PID    COMMAND          %CPU TIME       ...  USER      ...
38702  chrome           10.0 01:00.41   ...  johndoe   ...
38701  top              4.0  00:00.09   ...  johndoe   ...
38699  Spotify          3.0  01:00.07   ...  johndoe   ...
```

- The “top”, or “table of processes” command acts as a task manager and will show a table running processes and their resource usage. This command comes in handy when you need to monitor system performance or identify resource-heavy processes. Here we show a usage example with the “minus n” option and the number “3” to display the top three running tasks: Chrome, top, and Spotify. By default, tasks are sorted by CPU usage. This is a simplified output of the “top” command for this video, but “top” provides many other details such as memory usage and the executable file location.



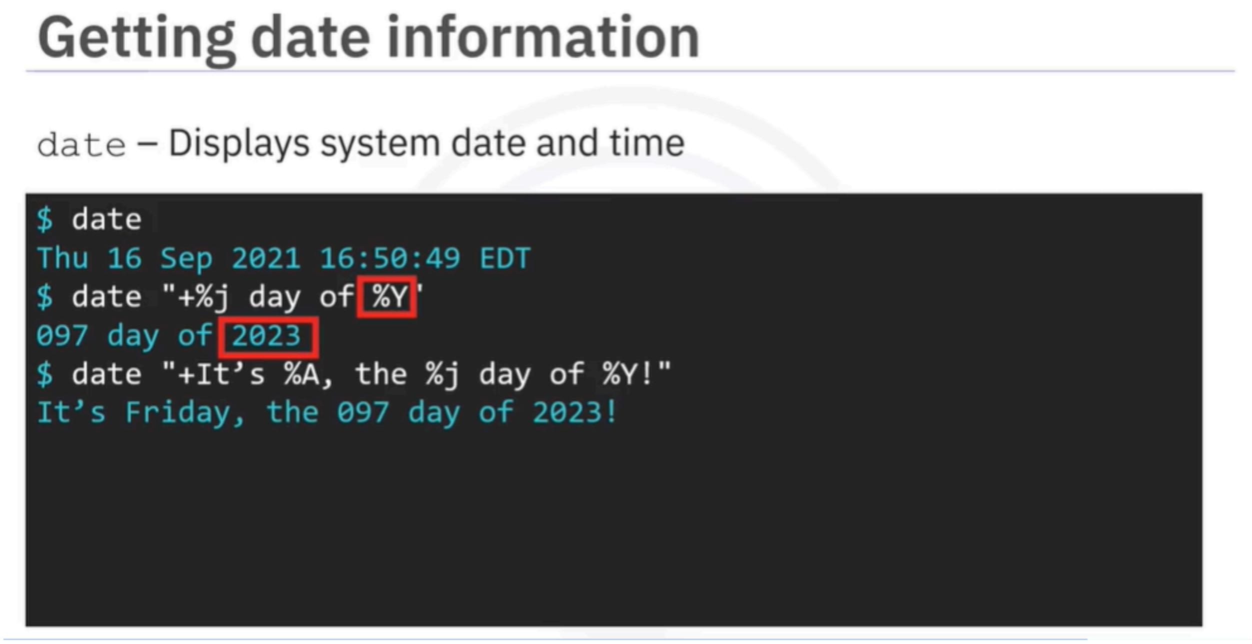
Printing strings and variable values

echo – Print string or variable value

```
$ echo

$ echo Hello
Hello
$ echo "Learning Linux is fun!"
Learning Linux is fun!
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

- Although simple, the "echo" command in Linux is a powerful tool to display text or variables on the terminal or in a shell script. Entering just “echo” is similar to telling the terminal to print nothing, and it will return a newline. If you would like to echo a single word such as “hello”, you can simply type “echo hello”, and the terminal will return the word. Strictly speaking, you don't need to add quotes around a string with spaces for echo to work as expected, but it's considered best practice to include quotes. Using echo with a quoted string returns the quoted contents, “Learning Linux is fun!" Additionally, you can view the value of a variable, such as our system’s “PATH” variable, by typing a dollar sign followed by the variable name. This can be helpful when troubleshooting or scripting. Here you can see each path of your system’s PATH variable, delimited by colons.

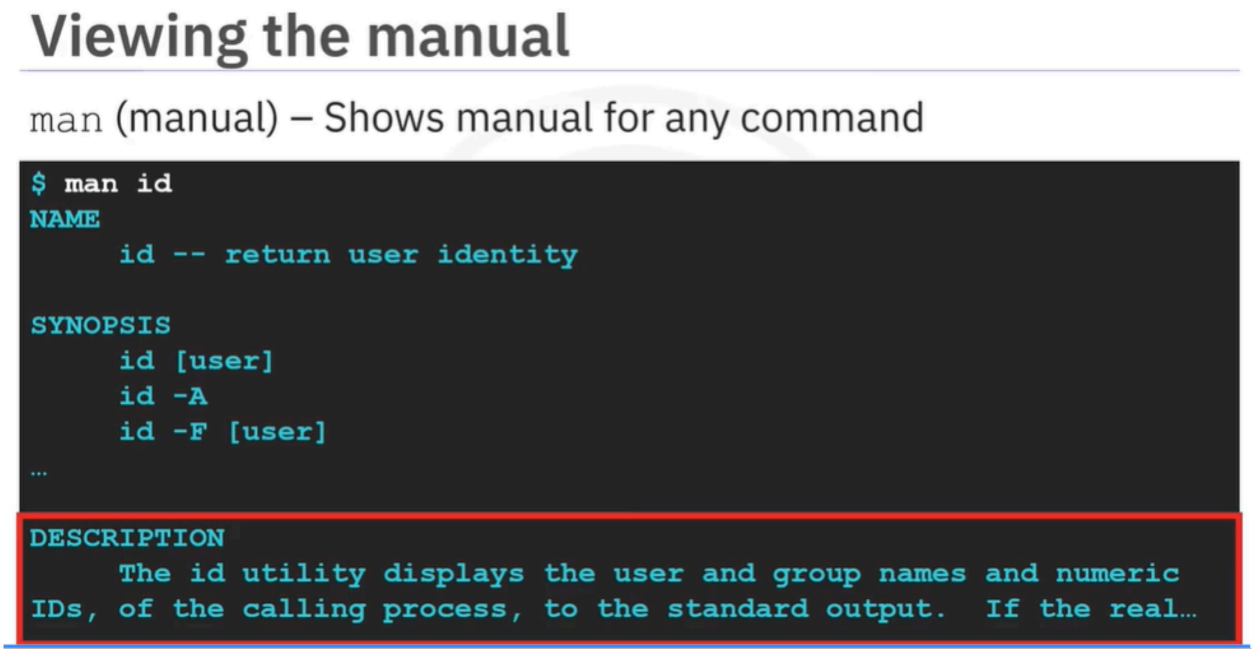


Getting date information

date – Displays system date and time

```
$ date
Thu 16 Sep 2021 16:50:49 EDT
$ date "+%j day of %Y'
097 day of 2023
$ date "+It’s %A, the %j day of %Y!"
It’s Friday, the 097 day of 2023!
```

- Another helpful command is “date,” which displays the current system date and time. Entering “date” returns the default date format: the day of the week, the day, month, year, time, and the time zone. You can also extract certain parts of the date to print, like in this example. To format the output, encapsulate within quotes a combination of text and control characters preceded by a “plus” sign. Format controls are indicated with the % symbol. In this case, “percent j” and “percent Y” output the numerical day of the year and the year itself, respectively. This command prints “97”, for the 97thth day of the year, the words “day of”, and the year, “2023”. Notice how “percent Y” is replaced by the year 2023. Finally, here is another example to show how you can further combine format controls with text to print unique strings. By bringing together controls like “percent A”, “percent j,” and “percent Y” with text, you can print a line that tells you the day of the week, the day, and the year.



Viewing the manual

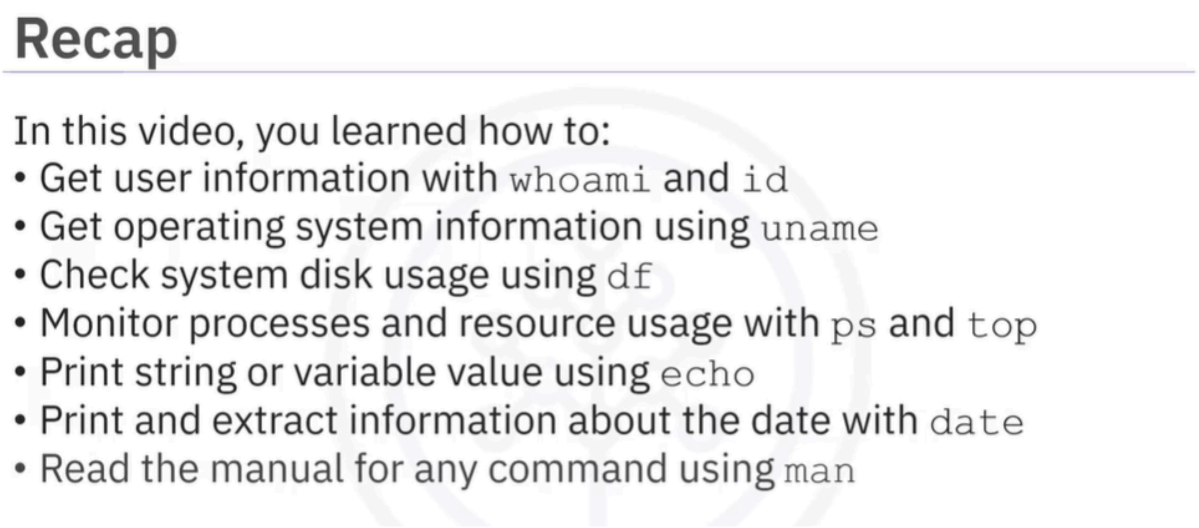
`man` (manual) – Shows manual for any command

```
$ man id
NAME
     id -- return user identity

SYNOPSIS
     id [user]
     id -A
     id -F [user]
…

DESCRIPTION
     The id utility displays the user and group names and numeric
IDs, of the calling process, to the standard output.  If the real…
```

- This video has only covered the basics of these commands. If you want to learn more about how to use a command, you can use the “man”, or “manual,” command. All default Linux commands come with a manual that you can display using “man.” For example, entering “man ID” will display the manual for the “ID” command. The manual will provide a basic summary of what the command does, such as “return user identity”. Also listed are the options for the “ID” command, such as “minus a.” Square brackets indicate optional parameters, such as “user,” which allows you to specify a username. The “man” command also provides a more detailed description of the command, which explains it in greater detail. The “man” command even has its own “man” page—you can use it to learn more about the manual command and its uses.



Recap

In this video, you learned how to:

- Get user information with `whoami` and `id`
- Get operating system information using `uname`
- Check system disk usage using `df`
- Monitor processes and resource usage with `ps` and `top`
- Print string or variable value using `echo`
- Print and extract information about the date with `date`
- Read the manual for any command using `man`