LINUX PROGRAMMING

ASSIGNMENT-10

NAME: S.NIVETHA
USN NO:ENG24CY0191
ROLL NO:63
CLASS:CY_3

history -c: Clear Command History

This command relates to the Bash (Bourne-Again Shell) history mechanism, which tracks commands executed by a user.

(1) What is the said command (little history, invention, authors of the command)?

history Command: The history command is an internal command of the Bash shell, which was written by Brian Fox and Chet Ramey in the late 1980s/early 1990s as a free software replacement for the proprietary Bourne Shell. Command history is a core feature of many modern shells (like Bash, Zsh, etc.).

Purpose: The history command lists commands previously entered by the user. The command history is typically stored in a file (e.g., ~/.bash_history).

history -c: This is the specific usage (with the -c flag) that clears the history list *in memory* for the current shell session. It does not automatically clear the persistent history file on disk unless you immediately save the history afterward (e.g., with history -w).

(2) Why the particular command is useful for System administrator?

Security and Privacy: Clearing the history is crucial if a System Administrator (SysAdmin) executes commands containing sensitive information like passwords, API keys, or private data (e.g., in a script or directly on the command line). Running history -c (and then potentially history -w to write the cleared memory to the file) helps prevent others from finding this sensitive data by checking the history file.

System Auditing and Cleanliness: In some cases, a SysAdmin may want to clear history to ensure that only commands executed *after* a certain point are recorded, perhaps for a specific, isolated task or an audit trail.

(3) How these said commands are working? Give proper example.

The history command maintains two histories: one in the shell's memory (the current session's history) and one in the history file (usually ~/.bash_history).

How history -c Works: It deletes all entries from the current shell session's in-memory history list.

Example (Working):

1. Check History: Run history | tail -3 to see the last three commands.
2. Clear In-Memory History:

Bash

```
$ history -c
```

3. Verify (History is Cleared in Session):

Bash

```
$ history
1  history
2  history -c
3  history
```

(Output will only show the three commands just executed after the clear.)

4. *(Optional: To clear the disk file as well, you would run:* history -w *after* history -c *to write the empty in-memory history to the file.)*

(4)Few snapshots of the command execution with proper illustration.

Step	Command	Output/Action
Initial State	`history	tail -3`
Clear In-Memory	$ history -c	*(No output)*
Check New State	$ history	1 history -c $\newline$ 2 history

(5) At least elaborate five to six Option and Flags related to the said command.

Option/Flag	Description
-c	Clears the history list in memory (the current session's list).
-r	Reads the contents of the history file and appends them to the current history list in memory.

Option/Flag	Description
-w	Writes the current history list (in memory) to the history file, overwriting the file's contents.
-a	Appends the new history lines (those entered in the current session since the last write/read) to the history file.
-n	Reads history lines not already read from the history file and appends them to the current history list.
-d OFFSET	Deletes the history entry at the specified $OFFSET$ (the line number) from the history list.

Source for detailed flags and options: Use the built-in manual page: $ help history or $ man bash and search for the history section.

virsh: Manage Virtual Machines

The virsh command is the main command-line utility for managing virtual machines (VMs) and hypervisors via the libvirt virtualization API.

(1) What is the said command (little history, invention, authors of the command)?

virsh Command: It is the Virtual Shell utility, which acts as a command-line interface to the libvirt virtualization API.

History/Author: libvirt was originally developed by Daniel P. Berrangé and others, primarily for the Xen hypervisor, and later expanded to support various hypervisors like KVM, VMware ESX/GSX, Microsoft Hyper-V, etc. libvirt began around 2005 and is a key component in the Linux virtualization ecosystem. virsh is the principal tool provided by the libvirt project.

Purpose: It allows for comprehensive management of virtual guests, networks, and storage on a host system.

(2) Why the particular command is useful for System administrator?

Centralized Management: virsh provides a single, consistent interface for managing multiple different hypervisors (KVM, Xen, etc.).

Automation: As a command-line tool, it is essential for writing scripts and automation tools (like Ansible, Puppet, etc.) to provision, start, stop, and manage VMs at scale.

Remote Administration: SysAdmins can use virsh over SSH to securely manage VMs on remote hosts without needing a graphical console (like *virt-manager*).

Troubleshooting: It allows for quick checking of VM status, network configuration, and live resource adjustments.

-

(3) How these said commands are working? Give proper example.

The virsh command works by communicating with the libvirt daemon (libvirtd) running on the host system. The daemon abstracts the underlying hypervisor (like KVM) and executes the requested action.

How virsh Works: A virsh command uses a subcommand to perform a specific action (e.g., list, start, shutdown), often requiring a VM name or ID (called a "domain" in libvirt terminology).

Example (Working):

1. List Running VMs:

Bash

```
$ virsh list
 Id   Name                         State
 ---------------------------------------------------
  1    production_web_server         running
```

2. Gracefully Shut Down a VM:

Bash

```
$ virsh shutdown production_web_server
Domain 'production_web_server' is being shutdown
```

3. Verify Status (Will show 'shut off'):

Bash

```
$ virsh list --all
 Id    Name                        State
----------------------------------------------------
 -     production_web_server         shut off
 -     dev_test_vm                 shut off
```

(4) Few snapshots of the command execution with proper illustration.

Step	Command	Output/Action
List All VMs	$ virsh list --all	Id Name State $\newline$ --- $\newline$ 4 linux_client_01 running $\newline$ - windows_server_01 shut off
Start a VM	$ virsh start windows_server_01	Domain 'windows_server_01' started
Get VM Details	$ virsh dominfo linux_client_01	Id: 4 $\newline$ Name: linux_client_01 $\newline$ UUID: 7d3a... $\newline$ ...State: running

(5) At least elaborate five to six Option and Flags related to the said command.

Option/Flag (Subcommand)	Description
list [--all]	Lists domains (VMs). --all shows active and inactive (shut off) domains.
start DOMAIN	Starts a defined (inactive) domain/VM.
shutdown DOMAIN	Performs a graceful shutdown of a running domain.
destroy DOMAIN	Forces an immediate, ungraceful power-off of a running domain. Use with caution.
reboot DOMAIN	Reboots a running domain gracefully.
dominfo DOMAIN	Provides detailed information about a specific domain (ID, UUID, state, memory).

Option/Flag (Subcommand)	Description
domifaddr DOMAIN	Shows the IP addresses and MAC addresses of a domain's network interfaces.

-