|  |
| --- |
| *Agnel Charities*  **Fr. C. Rodrigues Institute of Technology, Vashi**  **Department of Electronics and Telecommunication Engg.**  **SUB:-Linux & Networking & Server Configuration ( LNSC) ECL-604** |
| **Administrative functionality-1: System Admin Commands** |

|  |  |
| --- | --- |
| **Experiment No** | 04 |
| **AIM** | **Administrative functionality-1: System Admin Commands** |
| **THEORY** | Work of [system admin](https://geekflare.com/linux-books-videos/) includes installing and running software, controlling access, monitoring, ensuring availability, backups, restoring backups, etc. Some of the commands frequently used by Linux system administrators in day to day work are studied in the laboratory. |
| |  |  | | --- | --- | | **who and w, whoami** | **w** | | who shows users who are logged on.  ubuntu@ubuntu18:~$ who  ubuntu pts/0 2020-08-14 17:28 (183.83.211.129)  ubuntu pts/1 2020-08-14 17:58 (183.83.211.129) | It shows users currently logged on and their processes. The header shows the current time, system uptime, number of users logged on, and system load averages.  ubuntu@ubuntu18:~$ w  18:07:33 up 46 days, 15:19, 2 users, load average: 0.00, 0.00, 0.00  USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT  ubuntu pts/0 183.83.211.129 17:28 2.00s 0.10s 0.00s w  ubuntu pts/1 183.83.211.129 17:58 9:07 0.05s 0.01s vi |      |  |  | | --- | --- | | **uname** | **df** | | Use uname command with the -a flag to print system information. This command will show you the kernel name, kernel release, kernel version, hostname, processor type & your hardware platform information.  ubuntu@ubuntu18:~$ uname -a  Linux ubuntu18 5.3.0-1028-azure #29~18.04.1-Ubuntu SMP Fri Jun 5 14:32:34 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux  Here is what this means:  kernel name:Linux  hostname: ubuntu18  kernel release: 5.3.0-1028-azure  kernel version: #29~18.04.1-Ubuntu SMP Fri Jun 5 14:32:34 UTC 2020  machine hardware name: x86\_64  processor: x86\_64  hardware-platform: x86\_64  operating system: GNU/Linux | Use df command to verify the size of the filesystem and the space available. This command used by itself shows output in 1K blocks.  ubuntu@ubuntu18:~$ df  Filesystem 1K-blocks Used Available Use% Mounted on  udev 437208 0 437208 0% /dev  tmpfs 91100 692 90408 1% /run  /dev/sda1 30309264 2383952 27908928 8% / |   **du**  To check disk space usage in a directory, use du command. For example to see disk space usage in the /var/log directory. Use -h flag for human-readable format.  ubuntu@ubuntu18:~$ sudo du -h /var/log  24K /var/log/Microsoft/Azure/NetworkWatcherAgent/Logs  28K /var/log/Microsoft/Azure/NetworkWatcherAgent  32K /var/log/Microsoft/Azure    **free**  Use free command to see total, used, and free system memory. Use -h flag for human-readable format.  ubuntu@ubuntu18:~$ free -h  total used free shared buff/cache available  Mem: 889M 272M 100M 712K 517M 443M  Swap: 0B 0B 0B  total - Total installed memory (memtotal + swaptotal)  used - used memory  free - unused memory (memfree + swapfree)  buffers - memory used by kernel buffers  cache - memory used by page caches  buff/cache - sum of buffers and cache  available - Estimated memory available for starting new applications, without swapping  **ps**  Use ps to display status information about processes running on the system. To see all processes owned by user ubuntu, use -u flag with the user name:  ubuntu@ubuntu18:~$ ps -u ubuntu  PID TTY TIME CMD  7804 ? 00:00:00 systemd  7805 ? 00:00:00 (sd-pam)  7940 ? 00:00:00 sshd  7941 pts/0 00:00:00 bash  8111 ? 00:00:00 sshd  **top**  While ps command shows a snapshot of the state of processes at any moment, top shows a continuously updating (every three seconds, by default) list of system processes in order of process activity.  The top command output consists of two main parts: The system summary at the top and the table of processes sorted by CPU activity.  top - 14:25:32 up 44 days, 11:37, 1 user, load average: 0.00, 0.00, 0.00  Tasks: 114 total, 1 running, 59 sleeping, 0 stopped, 0 zombie  %Cpu(s): 0.3 us, 0.0 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  KiB Mem : 910992 total, 101208 free, 274712 used, 535072 buff/cache  KiB Swap: 0 total, 0 free, 0 used. 458492 avail Mem  PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  50497 ubuntu 20 0 44528 3944 3368 R 0.7 0.4 0:00.15 top  1 root 20 0 160076 7020 4400 S 0.0 0.8 0:34.85 systemd  2 root 20 0 0 0 0 S 0.0 0.0 0:00.08 kthreadd  3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu\_gp  4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu\_par\_gp  6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:+  9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm\_percpu\_+  **dig**  dig is a great tool for DNS queries. It is used as follows :  dig <DNS server> <domain> <query-type>  where   * <DNS server> is the DNS server name you wish to query * <domain> is the domain name you wish to query about * <query-type> is the name of the record you wish to know – A, MX, NS SOA, etc.   To suppress verbose output, use +short flag.  **tar**  With GNU tar you can archive multiple files into a single file.  As an example create a directory myfiles and three files a.txt, b.txt, c.txt in myfiles directory:  ubuntu@ubuntu18:~$ mkdir myfiles ; touch myfiles/{a.txt,b.txt,c.txt}  Now to create an archive named allfiles.tar containing all files in myfiles directory:  ubuntu@ubuntu18:~$ tar -cvf allfiles.tar myfiles  myfiles/  myfiles/c.txt  myfiles/a.txt  myfiles/b.txt  List all files in the current directory. You can see myfiles directory and allfiles.tar archive:  ubuntu@ubuntu18:~$ ls  allfiles.tar myfiles  **grep**  grep is used to search for a pattern in a file, or a set of files. It print all lines matching that pattern. For example, to search for the line containing “ServerRoot” in /etc/apache2/apache2.conf:  ubuntu@ubuntu18:~$ grep ServerRoot /etc/apache2/apache2.conf  # ServerRoot: The top of the directory tree under which the server's  #ServerRoot "/etc/apache2"  To search in all files in a directory use \*. To include search in subdirectories use -r (recursive) flag. So, to search for all lines containing the pattern “VirtualHost” in all files in /etc/apache2:  ubuntu@ubuntu18:~$ cd /etc/apache2  ubuntu@ubuntu18:/etc/apache2$ grep -r VirtualHost \*  apache2.conf:# If you do not specify an ErrorLog directive within a <VirtualHost>  apache2.conf:# logged here. If you \*do\* define an error logfile for a <VirtualHost>  conf-available/localized-error-pages.conf:# even on a per-VirtualHost basis. If you include the Alias in the global server  conf-available/other-vhosts-access-log.conf:# Define an access log for VirtualHosts that don't define their own logfile  ports.conf:# have to change the VirtualHost statement in  sites-available/000-default.conf:<VirtualHost \*:80>  try awk command  **ss**  ss command is used to dump socket statistics, similar to the legacy netstat utility. To display TCP sockets use -t flag.  ubuntu@ubuntu18:~$ ss -t  State Recv-Q Send-Q Local Address:Port Peer Address:Port  ESTAB 0 0 10.0.0.4:53852 168.63.129.16:8037  ESTAB 0 0 10.0.0.4:ssh 183.83.211.129:64118  ESTAB 0 0 10.0.0.4:33256 169.254.169.254:http  ESTAB 0 1080 10.0.0.4:ssh 222.186.30.35:11527  ESTAB 0 0 10.0.0.4:ssh 183.83.211.129:63049  This would not display sockets that are listening. To include both listening and non-listening sockets use -t and -a flags.  ubuntu@ubuntu18:~$ ss -t -a  State Recv-Q Send-Q Local Address:Port Peer Address:Port  LISTEN 0 128 0.0.0.0:ssh 0.0.0.0:\*  LISTEN 0 80 127.0.0.1:mysql 0.0.0.0:\*  LISTEN 0 128 127.0.0.53%lo:domain 0.0.0.0:\*  ESTAB 0 0 10.0.0.4:53852 168.63.129.16:8037  ESTAB 0 0 10.0.0.4:ssh 183.83.211.129:64118  ESTAB 0 0 10.0.0.4:33256 169.254.169.254:http  ESTAB 0 1080 10.0.0.4:ssh 222.186.30.35:11527  ESTAB 0 120 10.0.0.4:ssh 183.83.211.129:63049  LISTEN 0 128 [::]:ssh [::]:\*  LISTEN 0 128 \*:http \*:\*  **locate**  The locate command uses a database to search for files and actually can be much faster than find command. Very simple to use, to search for a file, say, apache2.conf:  ubuntu@ubuntu18:~$ locate apache2.conf  /etc/apache2/apache2.conf  /var/lib/dpkg/info/apache2.conffiles  You can use -c flag if you want only the count of files matching the search pattern.  ubuntu@ubuntu18:~$ locate -c apache2.conf  2  At times, you may need to refresh the database used by locate, which is mlocate. To update the database use updatedb command. This would need superuser privileges.  ubuntu@ubuntu18:~$ sudo updated  **find**  One of the most frequently used commands on Linux. Use it to search for files based on filenames, permissions, userid,  groupid, size, file type, besides other criteria.  To search for a file by name in the current directory, use -name flag followed by the filename to search.:  ubuntu@ubuntu18:~$ find . -name a.txt  ./myfiles/a.txt  To search for directories, use -type d flag:  ubuntu@ubuntu18:~$ find . -type d  .  ./.ssh  ./myfiles  ./.cache  ./.gnupg  ./.gnupg/private-keys-v1.d  ./docker  To search for files by size, say files larger than 20MB, use -size flag:  ubuntu@ubuntu18:~$ find . -size +20M  ./docker/docker-ce-cli\_5%3a19.03.12~3-0~ubuntu-bionic\_amd64.deb  ./docker/docker-ce\_5%3a19.03.12~3-0~ubuntu-bionic\_amd64.deb  **systemctl**  Now that systemd has replaced SysV init process in most Linux distributions, use [systemctl](https://geekflare.com/manage-systemd-services-with-systemctl/) command to manage systemd services and units.  To start , stop , status a service, for example apache2:  ubuntu@ubuntu18:~$ sudo systemctl start apache2.service  To stop a service:  ubuntu@ubuntu18:~$ sudo systemctl stop apache2  To see service status, use systemctl status command:  ubuntu@ubuntu18:~$ sudo systemctl status apache2  **ufw command**  UFW – uncomplicated firewall is an easy to use frontend for iptables. It is available by default, on Ubuntu-based distributions. On CentOS, you can install ufw from the EPEL repository.  To enable ufw:  $ sudo ufw enable  Check firewall status with ufw status:  $ sudo ufw status  Status: active  Default UFW policies allow all outgoing traffic and block all incoming traffic.  The following command allows incoming  traffic on HTTP port:  $ sudo ufw allow http  Rule added  Rule added (v6)  You can deny traffic on any port. Here is an example to block traffic on port 21:  $ sudo ufw deny 21  Rule added  Rule added (v6)  **kill and killall**  You may need to kill a runaway process or when you need to free some system resources. kill with -l flag shows all the signals you can send to a process.  ubuntu@ubuntu18:~$ kill -l  1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP  6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1  11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM  16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP  ....  To kill apache2 process with process id 45525:  ubuntu@ubuntu18:~$ sudo kill -9 45525  Again view the list of apache2 processes:  ubuntu@ubuntu18:~$ ps aux|grep apache2  root 45521 0.0 0.5 78188 4688 ? Ss Aug08 0:03 /usr/sbin/apache2 -k start  www-data 45524 0.0 0.5 830480 4816 ? Sl Aug08 0:00 /usr/sbin/apache2 -k start  ubuntu 70525 0.0 0.1 14852 1052 pts/0 S+ 07:52 0:00 grep --color=auto apache2  Use killall to kill a program by name. Which means killall kills the control(parent) process and all child processes. To kill all instances of the apache2 process in the example above:  ubuntu@ubuntu18:~$ sudo killall apache2  Use kill and killall with caution. These commands might leave the system in an unstable state.  **IP**  The ip command replaces ifconfig in the newer Linux distributions. Use it to configure and display network interfaces. Also used to display and modify IP addresses, routes, and neighbor objects.  Display information about all network interfaces:  ubuntu@ubuntu18:~$ ip a  1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  inet 127.0.0.1/8 scope host lo  valid\_lft forever preferred\_lft forever  inet6 ::1/128 scope host  valid\_lft forever preferred\_lft forever  **date**  A very commonly used command, here we explore some interesting ways to use this command.  ubuntu@ubuntu18:~$ date  Tue Aug 11 07:42:49 UTC 2020  We can easily control date output format. Here is an example:  ubuntu@ubuntu18:~$ date '+%d-%B-%Y'  11-August-2020 | |
|  |  |
| **CONCLUSI-ON** | Main task of system administrator is to keep the system and [computing infrastructure](https://geekflare.com/cloud-computing-basics/) running, by solving the problems, maintain. System admin related required commands need to be practiced to keep the systems running optimally while ensuring security. |