|  |  |
| --- | --- |
| Linux Handbook |  |
| |  |  | | --- | --- | |  | **My very own LINUX Handbook** |  |  |  |  | | --- | --- | --- | |  | /bin ->Binaries  /sbin-> System Binaries  /boot  /dev  /etc  /lib /lib32 /lib64  /media /mnt  /opt  /proc  /home  /root  /run->Run directory  /snap  /srv-> Service directory  /sys-> System directory  /tmp  /urs  /var-> Variable directory | programs or applications ( ls , cat etc)  similar to bin, but accessable by root  Boot loader and other booting stuff  Hardware files (mouse keyboard etc…)  Configuration files (systemwide)  Stores libraries  Mount storage devices media-> automatic ; mnt-> manual  Optionals (manual installed softwares)  Sudo and process directories  private workspacefor storing personal files and more stuff  Root users’s home directory  runs in ram store runtime information  Contain snap(self run) packages mainly used by ubuntu  servisce data is stored here  similar to run directory  Temporary copy of running task is saved here  User application space  stores filed/directories which in future can grow in size | |  | ls  ls -lS  ls -lSr  lsblk  df -h | Shows all files and folders (directories)  Sort by file size  Reverse sort  List storage in tree format  List memory of system in human redable format | |  | cd  cd /  cd .. | Change directory (folders)  Switch to root  Switch one directory back | |  | Sudo  su | Substitute User Do  Switch user | |  | mkdir | Make new directory | |  | sh | Switch to “sh” shell | |  | uptime | Uptime of system | |  | w | Shows currently logged-in user | |  | which<file\_name> | Gives location of executable file | |  | man <command\_name> | Get help with the command | |  | <command> --help | Gets help with the command | |  | rm <file\_name> | Removes files | |  | rm -r <directory\_name> | Removes directories | |  | ps -a  ps aux  kill <PID> | Shows all currently running proceses  Shows all diff. running processes  Kills process for that corresponding pid | |  | systemctl list -units –type=service –state =running  systemctl cat “service\_name”  systemctl edit “service\_name” – full  cd /usr/bin  nano dummy.sh :  #!/bin/zsh  PATH=“/bin:/sbin:/usr/bin:/usr/sbin”  While true  Do  echo “Current Date & Time : $(date)”  sleep 3  done  chmod +x dummy.sh  /etc/systemd/system  Nano dummy.sh :  [Unit]  Description = first service  [Servise]  Execstart=/usr/sbin/dummy.sh  [Install]  wantedBy=multi\_user.target  systemctl enable dummy | List service units which are currently running  Read the service file  To edit service file  Create new service here  Dummy file for service  Done creating the service  Provide executable permission to file  Create unit file here for dummy.sh  Dummy enabled | |  | htop | Lists all processes running | |  | cat<file\_name> | Read contents of file | |  | pwd | Print working directory | |  | Data Streams : STDIN  STDOUT  STDERR | Input given Numeric Id: 0  Output Given Numeric Id: 0  Error output Numeric Id: 2 | |  | <command> 0 > file.txt  <command> 1 > file.txt  <command> 2 > file.txt | Sends input to file.txt  Sends output of STDOUT data stream to file.txt  Sends output of STDERR data stream to file.txt | |  | touch <File\_name> | Create an empty file | |  | Echo “text to be stored” > file\_name | Create a file with that text | |  | Vim <file\_name>  Shift+i  Esc  :wq | Open file with vim editor  Start editting  Exit editting mode  Written and quit | |  | nano <file\_name>  ctrl+x | Open file in nano editor. Can edit directly  Exit | |  | cp <file1> <file2>  cp <file path> <new\_file>  cp -r <directory\_path> <new dir name> | Copy content of file1 to file2  Copy content of file at that location to new\_file  Copy whole directory | |  | Mv <file\_name> <new\_name> | Rename a file | |  | head <file\_name>  head -n x <file\_name>  tail <file\_name>  tail -n x <file\_name>  more <file\_name> | Shows first 10 lines of the file  Shows first x lines of the file  Shows last 10 lines of the file  Shows last x lines of the file  Read whole file | |  | ls -l |grep <name>  cat <file\_name> | grep<text>  grep “text ro be searched” <file\_name>  grep ^“Alphabet”  grep $“Alphabet” | Filter the given name  Filter that text in the given file  -----------“---------------“-----------  Find words starting with that alphabet  Find words ending with that alphabet | |  | grep “\->”  ln -s <existing\_file> link\_name  ln < exixting\_file > link\_name | Filter links  Create a soft link  Create a hard link | |  | find -name <file\_name>  find -name “\*.txt”  find -name “\*.pdf”  find -type f  find -type d  find -type l  find -type [f/d/l] -ls  find -type [f/d/l] -empty  find / -name kernel -type d  find / -name Kernel -type d | xargs ls -l --color | Gives path to file  Will give all txt files  Will give all pdf files  Will give files  Will give directories  Will give links  Will show file/dict/links in tabulaar form  Will search for empty files/dict/links  Will show directories which include kernel in name  Will show directories which include kernel in name in colors | |  | ls | xargs cat | Show contents of every single file given by ls | |  | tar -cf file\_name.tar ~  tar -tf file\_name.tar  tar -czf filename.tar.gz | Create an archive of home directory  Reat an archive  Create a compressed zip file | |  | gzip file\_name | File will get compressed | |  | sort <file­\_name>  sort -r <file\_name>  sort -n <file\_name> | Give sorted contents of file  Sort in reverse  Sort integers | |  | useradd -m <username>  passwd username  adduser -m <username> (Better than Useradd)  su username  userdel username | Add new user with home dict in it  Set pass for user  Add new user with pre inbuilt dict and terminal will ask for info  Switch to new user  Delete user | |  | Groupadd <group\_name>  /etc/group  Sudo usermod –group group\_name -a user\_name | Create a new group  Location of all groups  Add user to group | |  | touch name.sh | Create an executable file with extention type “.sh” | |  | chmod  u->user ~ owner r->read + ->add  g->group w->write - ->remove  o->other users x->execute = ->execute  a->all  chmod u+ [r/w/x] usrer\_name  chmod u- [r/w/x] usrer\_name  chmod g+ [r/w/x] usrer\_name  chmod g- [r/w/x] usrer\_name  chmod o+ [r/w/x] usrer\_name  chmod o- [r/w/x] usrer\_name  chmod 777 <file\_name> | Change permissions u g o  r->4 rw->6 rwx rwx rwx  w->2 rx->5 421 421 421  x->1 wx->3 7 7 7  give user the permission to r/w/x  prohibit the user to r/w/x  give group the permission to r/w/x  prohibit the group to r/w/x  give other users the permission to r/w/x  prohibit the other users to r/w/x  give r , w & x to all 3 i.e., user , group & others | |  | chown <new\_owner\_name> file\_name  chgrp <new\_grp\_name> file\_name  OR  chown <new\_owner>:<new\_grp> file\_name | Change ownership of file  Change group of file  Change both owner and grp | |  | hostname  hostname -i  ip -4 a  ip -6 a | Shows hostname of system  Shows ip address of system  Shows Ipv4 address of system  Shows Ipv6 address of system | |  | ping <server>  ping -4 <server> | Verifies connectivity with server [Ipv4]  Verifies connectivity with server [Ipv6] | |  | nslookup <server>  whois <server> | for manual DNS lookups  Gives owner of server | |  | Ssh  service ssh status  service ssh start  service ssh enable  service ssh stop  service ssh disable  ssh username@ipAddress  ssh -p [newPort no.] username@ipAdress  nano /etc/ssh/sshd\_config  sshd -t  Any custom msg can be displayed with ssh login | Secure Shell  Check status of ssh service  Start ssh service  Enable ssh service  Stop ssh service  disable ssh service  Remotely connect to ssh with another pc with default port  Remotely connect to ssh with custom port number  File to edit logs of ssh for root and many more  Check errors in sshd\_config  ->Create a file in /etc  ->mounting it to banner in sshd\_config | |  | curl <webpage\_name>  curl url > file\_name  wget url | Get html content of the page  Save html of page to file  Will download the item | |  | env  export <variable\_name> = test | Get list of all environmental variables  Create new variable in test | |  | file <file\_name> | Shows info. about file type | |  | Apache3  Systemctl is-active apache3  systemctl status apache3  systemctl start apache  systemctl stop apache  systemctl enable apache  systemctl disable apache  systemctl restart apache | Web server  Check if apache3 or any other service is active  Check status for apache3 or any other service  start apache3 or any other service  stop apache3 or any other service  enable apache3 or any other service  dIsable status for apache3 or any other service  restart apache3 or any other service | |  | Journalctl  Journalctl -f  Journalctl -f -u ssh | For log entries  To folloe logs  To view ssh logs f->follow u->unit | |  | htop t->tcp  iotop u->udp  iptraf l->listning  ss -t -u -l | Monitoring system  Input and output monitoring  Monitor networks  List all listning ports | |  | docker run -it debian | Install debian container in wsl(Windows Sunsystem for Linux) | |  | cd / mnt  mkdir backups  mount dev / [sda or dsb or sdc] / mnt backups  nano / etc / fstab  /dev/ [ sda or sdb or sdc ] /mnt/backups ext 4 default 0 | Directory to add usb drive  Make backups directory  Mount usb in this path sda or sdb or sdc depends on system  Edit fsteb file to permanently mount usb  Create backup | |  | Linux Firewall = Netfilters + iptables  Tables->chains->rules  Rules = match + targets  iptables -L  iptables -t nat -L  iptables -nL  iptables -vL  iptables --line-numbers -L  iptables [-t table] -N CHAIN  iptables [-t table] -X CHAIN | Netfilters->kernel framework iptables->packet selection system  Displays the filter table  Displays the nat table  Displays using numeric output  Displays using vebros output  Use line nums  Create a new chain  Delete s preexisting chain | |  |  |  | |  |  |  | |  |  |  | |  |
|  |  |