1. **cut command**

**cut command can be used to cut parts of a line by delimiter, byte position, and character.**

### **Common Options:**

* **-d : Specifies the delimiter.**
* **-f : Specifies the field numbers to extract.**
* **-c : Specifies the character positions to extract.**
* **-b : Specifies the byte positions to extract.**

1. **Cut by byte position:**

**To extract specific byte positions**

**[root@ip-172-31-36-210 ~]# cat cutsample.txt**

**John Doe,30,Engineer**

**Jane Smith,25,Designer**

**Alice Johnson,28,Architect**

**Bob Brown,35,Manager**

**[root@ip-172-31-36-210 ~]# cut -b 1 cutsample.txt**

**J**

**J**

**A**

**B**

**[root@ip-172-31-36-210 ~]#**

**It’s extract first byte from file**

**[root@ip-172-31-36-210 ~]# cut -b 2 cutsample.txt**

**o**

**a**

**l**

**o**

**It’s extract second byte from file**

**[root@ip-172-31-36-210 ~]# cut -b 1-4 cutsample.txt**

**John**

**Jane**

**Alic**

**Bob**

**[root@ip-172-31-36-210 ~]#**

**This command extracts the first four bytes from each line in cutsample.txt**

1. **Cut by character:**

**Extract the first five characters from each line:**

**cut -c 1-5 cutsample.txt**

**This command extracts the first five characters from each line in cutsample.txt.**

**[root@ip-172-31-36-210 ~]# cut -c 1 cutsample.txt**

**J**

**J**

**A**

**B**

**[root@ip-172-31-36-210 ~]# cut -c 1-5 cutsample.txt**

**John**

**Jane**

**Alice**

**Bob B**

**[root@ip-172-31-36-210 ~]#**

**[root@ip-172-31-36-210 ~]# cut -c 1,4 cutsample.txt**

**Jn**

**Je**

**Ac**

**B**

**[root@ip-172-31-36-210 ~]#**

**1st and 4th character from line.**

### **Cut by delimiter and field:**

* **Extract the 1st and 3rd fields from each line, where fields are separated by a comma:  
    
  cut -d',' -f1,3 cutsample.txt**

**[root@ip-172-31-36-210 ~]# cut -d ‘,’ -f1,3 cutsample.txt**

**John Doe,Engineer**

**Jane Smith,Designer**

**Alice Johnson,Architect**

**Bob Brown,Manager**

**[root@ip-172-31-36-210 ~]#**

**[root@ip-172-31-36-210 ~]# cut -d ':' -f1,3,4 /etc/passwd**

**root:0:0**

**bin:1:1**

**daemon:2:2**

**adm:3:4**

**lp:4:7**

**sync:5:0**

**shutdown:6:0**

**halt:7:0**

**mail:8:12**

**operator:11:0**

**games:12:100**

**ftp:14:50**

**nobody:65534:65534**

**dbus:81:81**

**systemd-network:192:192**

**systemd-oom:999:999**

**systemd-resolve:193:193**

**sshd:74:74**

**rpc:32:32**

**libstoragemgmt:997:997**

**systemd-coredump:996:996**

**systemd-timesync:995:995**

**chrony:994:994**

**ec2-instance-connect:993:993**

**rpcuser:29:29**

**tcpdump:72:72**

**ec2-user:1000:1000**

**[root@ip-172-31-36-210 ~]**

**[root@ip-172-31-36-210 ~]# cut -d ':' -f1,3,4 /etc/passwd |cut -c 1-4**

**root**

**bin:**

**daem**

**adm:**

**lp:4**

**sync**

**shut**

**halt**

**mail**

**oper**

**game**

**ftp:**

**nobo**

**dbus**

**syst**

**syst**

**syst**

**sshd**

**rpc:**

**libs**

**syst**

**syst**

**chro**

**ec2-**

**rpcu**

**tcpd**

**ec2-**

**[root@ip-172-31-36-210 ~]#**

**If you want to replace the comma delimiter with a pipe (|) delimiter in the cut command, you can simply adjust the -d option .**

**[root@ip-172-31-36-210 ~]# cat cutsample.txt**

**John Doe,30,Engineer**

**Jane Smith,25,Designer**

**Alice Johnson,28,Architect**

**Bob Brown,35,Manager**

**[root@ip-172-31-36-210 ~]# cut -d ',' -f1- cutsample.txt --output-delimiter="|"**

**John Doe|30|Engineer**

**Jane Smith|25|Designer**

**Alice Johnson|28|Architect**

**Bob Brown|35|Manager**

**[root@ip-172-31-36-210 ~]#**

**Here 1- denotes from first field to last field**

**[root@ip-172-31-36-210 ~]# cut -d ',' -f1,3 cutsample.txt --output-delimiter="|"**

**John Doe|Engineer**

**Jane Smith|Designer**

**Alice Johnson|Architect**

**Bob Brown|Manager**

**[root@ip-172-31-36-210 ~]#**

**Or**

**Using tr command**

**[root@ip-172-31-36-210 ~]# cut -d ',' -f1,3 cutsample.txt |tr ',' '|'**

**John Doe|Engineer**

**Jane Smith|Designer**

**Alice Johnson|Architect**

**Bob Brown|Manager**

**[root@ip-172-31-36-210 ~]#**

**[root@ip-172-31-36-210 ~]# cut -d ':' -f1- /etc/passwd --output-delimiter=" | "**

**root | x | 0 | 0 | root | /root | /bin/bash**

**bin | x | 1 | 1 | bin | /bin | /sbin/nologin**

**daemon | x | 2 | 2 | daemon | /sbin | /sbin/nologin**

**adm | x | 3 | 4 | adm | /var/adm | /sbin/nologin**

**lp | x | 4 | 7 | lp | /var/spool/lpd | /sbin/nologin**

**sync | x | 5 | 0 | sync | /sbin | /bin/sync**

**shutdown | x | 6 | 0 | shutdown | /sbin | /sbin/shutdown**

**halt | x | 7 | 0 | halt | /sbin | /sbin/halt**

**mail | x | 8 | 12 | mail | /var/spool/mail | /sbin/nologin**

**operator | x | 11 | 0 | operator | /root | /sbin/nologin**

**games | x | 12 | 100 | games | /usr/games | /sbin/nologin**

**ftp | x | 14 | 50 | FTP User | /var/ftp | /sbin/nologin**

**nobody | x | 65534 | 65534 | Kernel Overflow User | / | /sbin/nologin**

**dbus | x | 81 | 81 | System message bus | / | /sbin/nologin**

**systemd-network | x | 192 | 192 | systemd Network Management | / | /usr/sbin/nologin**

**systemd-oom | x | 999 | 999 | systemd Userspace OOM Killer | / | /usr/sbin/nologin**

**systemd-resolve | x | 193 | 193 | systemd Resolver | / | /usr/sbin/nologin**

**sshd | x | 74 | 74 | Privilege-separated SSH | /usr/share/empty.sshd | /sbin/nologin**

**rpc | x | 32 | 32 | Rpcbind Daemon | /var/lib/rpcbind | /sbin/nologin**

**libstoragemgmt | x | 997 | 997 | daemon account for libstoragemgmt | / | /usr/sbin/nologin**

**systemd-coredump | x | 996 | 996 | systemd Core Dumper | / | /usr/sbin/nologin**

**systemd-timesync | x | 995 | 995 | systemd Time Synchronization | / | /usr/sbin/nologin**

**chrony | x | 994 | 994 | chrony system user | /var/lib/chrony | /sbin/nologin**

**ec2-instance-connect | x | 993 | 993 | | /home/ec2-instance-connect | /sbin/nologin**

**rpcuser | x | 29 | 29 | RPC Service User | /var/lib/nfs | /sbin/nologin**

**tcpdump | x | 72 | 72 | | / | /sbin/nologin**

**ec2-user | x | 1000 | 1000 | EC2 Default User | /home/ec2-user | /bin/bash**

**[root@ip-172-31-36-210 ~]# cut -d ':' -f1- /etc/passwd |tr ':' '|'**

**root|x|0|0|root|/root|/bin/bash**

**bin|x|1|1|bin|/bin|/sbin/nologin**

**daemon|x|2|2|daemon|/sbin|/sbin/nologin**

**adm|x|3|4|adm|/var/adm|/sbin/nologin**

**lp|x|4|7|lp|/var/spool/lpd|/sbin/nologin**

**sync|x|5|0|sync|/sbin|/bin/sync**

**shutdown|x|6|0|shutdown|/sbin|/sbin/shutdown**

**halt|x|7|0|halt|/sbin|/sbin/halt**

**mail|x|8|12|mail|/var/spool/mail|/sbin/nologin**

**operator|x|11|0|operator|/root|/sbin/nologin**

**games|x|12|100|games|/usr/games|/sbin/nologin**

**ftp|x|14|50|FTP User|/var/ftp|/sbin/nologin**

**nobody|x|65534|65534|Kernel Overflow User|/|/sbin/nologin**

**dbus|x|81|81|System message bus|/|/sbin/nologin**

**systemd-network|x|192|192|systemd Network Management|/|/usr/sbin/nologin**

**systemd-oom|x|999|999|systemd Userspace OOM Killer|/|/usr/sbin/nologin**

**systemd-resolve|x|193|193|systemd Resolver|/|/usr/sbin/nologin**

**sshd|x|74|74|Privilege-separated SSH|/usr/share/empty.sshd|/sbin/nologin**

**rpc|x|32|32|Rpcbind Daemon|/var/lib/rpcbind|/sbin/nologin**

**libstoragemgmt|x|997|997|daemon account for libstoragemgmt|/|/usr/sbin/nologin**

**systemd-coredump|x|996|996|systemd Core Dumper|/|/usr/sbin/nologin**

**systemd-timesync|x|995|995|systemd Time Synchronization|/|/usr/sbin/nologin**

**chrony|x|994|994|chrony system user|/var/lib/chrony|/sbin/nologin**

**ec2-instance-connect|x|993|993||/home/ec2-instance-connect|/sbin/nologin**

**rpcuser|x|29|29|RPC Service User|/var/lib/nfs|/sbin/nologin**

**tcpdump|x|72|72||/|/sbin/nologin**

**ec2-user|x|1000|1000|EC2 Default User|/home/ec2-user|/bin/bash**

**[root@ip-172-31-36-210 ~]#**

**Interview question**

1. **What is the difference between using the cut command with the -b (bytes) option and the -c (characters) option?**
2. **In which situations would using the -b option be appropriate, and when would you prefer the -c option?**
3. **Can you explain a scenario where using cut -b might produce different or unexpected results compared to cut -c? Please provide an example.**

**2) sed command**

**If you want to remove the # symbol from the beginning of the lines while keeping the rest of the line intact**

**[root@ip-172-31-36-210 ~]# cat sedexample.txt**

**#linux tutorial**

**linux is open-source. linux is very popular. linux is unix based os.**

**linux**

**linux tutorial.**

**www.linux.com .www.linux1.com**

**[root@ip-172-31-36-210 ~]# sed 's/^#//' sedexample.txt**

**linux tutorial**

**linux is open-source. linux is very popular. linux is unix based os.**

**linux**

**linux tutorial.**

**www.linux.com .www.linux1.com**

**[root@ip-172-31-36-210 ~]#**

### **Explanation:**

* **s/^#//: This command substitutes (removes) the # symbol at the beginning of a line (^ indicates the start of the line). It only removes the #, leaving the rest of the line as it is.**

**The caret symbol (^) has several uses depending on the context, such as in regular expressions, command-line tools, and programming.**

### **In sed Commands:**

* **Start of Line:**
  + **The ^ symbol is used in sed to match the start of a line, similar to regular expressions.**

**Example: To remove the # symbol from the beginning of lines:  
  
sed 's/^#//' file.txt**

* + **This command will remove # from the start of lines but leave the rest of the line intact.**

**[root@ip-172-31-36-210 ~]# cat sedexample.txt**

**#linux tutorial**

**linux is open-source. linux is very popular. linux is unix based os.**

**linux**

**linux tutorial.**

**www.linux.com .www.linux1.com**

**#the end . #hello world**

**[root@ip-172-31-36-210 ~]# sed 's/^#//' sedexample.txt**

**linux tutorial**

**linux is open-source. linux is very popular. linux is unix based os.**

**linux**

**linux tutorial.**

**www.linux.com .www.linux1.com**

**the end . #hello world**

**[root@ip-172-31-36-210 ~]#**

**Remove Lines Starting with a Specific Pattern:**

**To delete lines that start with "linux":**

**[root@ip-172-31-36-210 ~]# sed '/^linux/d' sedexample.txt**

**#linux tutorial**

**www.linux.com .www.linux1.com**

**#the end . #hello world**

**[root@ip-172-31-36-210 ~]#**

**In regular expressions, the caret symbol (^) has a specific meaning when used inside square brackets ([]). Here’s how it works:**

### **Caret (^) Inside Square Brackets ([])**

**When the caret (^) is placed at the beginning of a character class (inside the square brackets), it negates the class. This means it matches any character that is not listed in the character class.**

### **Examples:**

#### **1. Match Any Character Except Specific Ones:**

**Example: To match any character except digits:  
bash  
Copy code  
grep '[^0-9]' filename.txt**

* **Explanation: [^0-9] matches any character that is not a digit (0-9). This command will return lines from filename.txt that contain at least one non-digit character.**

**Example ➖**

**Remove Lines Containing Any Character Except Letters (a-z):**

**[root@ip-172-31-36-210 ~]# cat caretsymbol.txt**

**apple**

**banana**

**cherry**

**1234**

**a1b2c3**

**!@#$**

**[root@ip-172-31-36-210 ~]# sed '/[^a-z]/d' caretsymbol.txt**

**apple**

**banana**

**cherry**

**[root@ip-172-31-36-210 ~]#**

**Explanation:**

* **[^a-z] matches any character that is not a lowercase letter.**
* **The d command deletes lines that contain any character other than lowercase letters.**

**Remove Lines Containing Any Characters Except Digits:**

**[root@ip-172-31-36-210 ~]# sed '/[^0-9]/d' caretsymbol.txt**

**1234**

**[root@ip-172-31-36-210 ~]#**

**Explanation of Output:**

* **Only the line containing digits 1234 is kept; all other lines are deleted because they contain non-digit characters.**

**[root@ip-172-31-36-210 ~]# grep '[^a-z]' caretsymbol.txt**

**1234**

**a1b2c3**

**!@#$**

**[root@ip-172-31-36-210 ~]# grep '[^0-9]' caretsymbol.txt**

**apple**

**banana**

**cherry**

**a1b2c3**

**!@#$**

**[root@ip-172-31-36-210 ~]#**

**3) awk command**

**Question:**

***You have a hostname in the format ip-172-31-36-210.ap-south-1.compute.internal. How would you extract the IP address 172.31.36.210 using awk in a Linux shell? Provide a solution that ensures only the IP address is returned, without any additional parts of the hostname.***

***[root@ip-172-31-36-210 ~]# hostname***

***ip-172-31-36-210.ap-south-1.compute.internal***

***[root@ip-172-31-36-210 ~]# hostname | awk -F "-" '{print $2"."$3"."$4"."$5}' |awk -F "." '{print $1"."$2"."$3"."$4}'***

***172.31.36.210***

***[root@ip-172-31-36-210 ~]#***

***Using cut command***

***[root@ip-172-31-36-210 ~]# hostname***

***ip-172-31-36-210.ap-south-1.compute.internal***

***[root@ip-172-31-36-210 ~]# hostname |cut -c 4-16***

***172-31-36-210***

***[root@ip-172-31-36-210 ~]# hostname |cut -c 4-16 |tr '-' '.'***

***172.31.36.210***

***[root@ip-172-31-36-210 ~]#***

***Now add amazon.com in above ip address***

***[root@ip-172-31-36-210 ~]# extracted\_ip=$(hostname | awk -F "-" '{print $2"."$3"."$4"."$5}' |awk -F "." '{print $1"."$2"."$3"."$4}')***

***[root@ip-172-31-36-210 ~]# echo $extracted\_ip***

***172.31.36.210***

***[root@ip-172-31-36-210 ~]# echo "$extracted\_ip.amazon.com"***

***172.31.36.210.amazon.com***

***[root@ip-172-31-36-210 ~]#***

**Question:**

***How can you calculate the total size of files in a directory using awk?***

***​​Answer: You can use awk to sum up the file sizes by processing the output of ls -lrt. For example, ls -lrt | awk -F " " 'BEGIN{sum=0}{sum=sum+$5}END{print sum}' adds up the sizes from the fifth column and prints the total.***

**Question:**

***Counting Error Messages in a Log File***

***[root@ip-172-31-36-210 ~]# cat saplelog.txt***

***INFO: Server started at 10:00 AM***

***ERROR: Failed to connect to the database***

***INFO: User login successful***

***ERROR: Timeout while connecting to service***

***INFO: Server shut down at 6:00 PM***

***ERROR: Disk space low***

***INFO: Server started at 10:00 AM***

***ERROR: Failed to connect to the database***

***INFO: User login successful***

***ERROR: Timeout while connecting to service***

***INFO: Server shut down at 6:00 PM***

***ERROR: Disk space low***

***INFO: Server started at 10:00 AM***

***ERROR: Failed to connect to the database***

***INFO: User login successful***

***ERROR: Timeout while connecting to service***

***INFO: Server shut down at 6:00 PM***

***ERROR: Disk space low***

***INFO: Server started at 10:00 AM***

***ERROR: Failed to connect to the database***

***INFO: User login successful***

***ERROR: Timeout while connecting to service***

***INFO: Server shut down at 6:00 PM***

***ERROR: Disk space low***

***INFO: Server started at 10:00 AM***

***ERROR: Failed to connect to the database***

***INFO: User login successful***

***ERROR: Timeout while connecting to service***

***INFO: Server shut down at 6:00 PM***

***ERROR: Disk space low***

***[root@ip-172-31-36-210 ~]# awk 'BEGIN{count=0} /ERROR/ {count=count+1} END {print count}' saplelog.txt***

***15***

***[root@ip-172-31-36-210 ~]#***

**Question:**

***You want to count the number of users who have /bin/bash as their default shell.***

***[root@ip-172-31-36-210 ~]# awk -F":" '$7 == "/bin/bash" {count++} END {print count}' /etc/passwd***

***2***

***[root@ip-172-31-36-210 ~]#***

**Question:**

**[root@ip-172-31-36-210 ~]# cat students.txt**

**John 85**

**Alice 90**

**Bob 75**

**Eve 95**

**Chris 60**

***You want to print the names of students who scored more than 80.***

***[root@ip-172-31-36-210 ~]# awk '{if ($2 > 80) print $1}' students.txt***

***John***

***Alice***

***Eve***

***[root@ip-172-31-36-210 ~]#***

***Let's say you want to classify the students based on their scores as "Pass" or "Fail." A passing score is 70 or above.***

***[root@ip-172-31-36-210 ~]# awk '{if ($2 >= 70) print $1, "Pass"; else print $1, "Fail"}' students.txt***

***John Pass***

***Alice Pass***

***Bob Pass***

***Eve Pass***

***Chris Fail***

***[root@ip-172-31-36-210 ~]#***

**Question:**

**[root@ip-172-31-36-210 ~]# cat disk\_usage.txt**

**Filesystem Size Used Avail Use% Mounted on**

**/dev/sda1 50G 45G 5G 90% /**

**/dev/sda2 100G 30G 70G 30% /home**

**/dev/sdb1 20G 18G 2G 90% /var**

**/dev/sdc1 10G 1G 9G 10% /backup**

**[root@ip-172-31-36-210 ~]#**

**You want to identify and print the filesystems that have usage above 80%.**

**​​[root@ip-172-31-36-210 ~]# awk 'NR > 1 && $5 > 80 {print $1, "is at", $5 " usage"}' disk\_usage.txt**

**/dev/sda1 is at 90% usage**

**/dev/sdb1 is at 90% usage**

**[root@ip-172-31-36-210 ~]#**