The Basic Questions you need to be Prepared:-  
===========================================  
1. Tell about yourself?  
2. What are your current Roles and Responsibilities?  
3. How Much experience do you have?  
4. What was the flow of Software development?  
5. What tools are you aware of?  
6. Have you worked on LINUX?  
7. Make an overview of tools that will be used in Build and Release?  
8. List out Version control commands?  
9. Will update more......  
  
  
Technical Questions:-  
===================  
1. What are the version controls you have used?  
2. Have you worked on any Continuous Integration Tools?

**3. Do you have any experience with scripting languages?**Sir if we talk from the time point of view then I have a little experience of only 1 year but I have made a lot of scripts.

**4. Do you know shell scripting?**A shell script is a computer program designed to be run by the unix shell, the command line interpreter. There are various operations performed by the shell scripts: file manipulation, program execution, and printing text.

**5. What software do you use for Installers?**

**6. Have you written any build files using ANT/NANT?**

**7. Linux Commands-->df,du,ps,umount,awk,time,ls,grep,top,head**df command is used to check the disk space.

df –k (in kilobytes) , df –h will display in MB,GB

**8. How to create new user in LINUX?**if we want to add new users in linux we can use the command useradd with a lot of options as per requirements:

Here we will take username as tecmint and practice on it.

1. **useradd tecmint**

This will create a user sahil. Now if want to unlock this user we will give the commad as:

**passwd tecmint**

then we can give the new password and confirm it.

**Once the new user is created its entry is automatically added to /etc/passwd file.**

The entry of the user in passwd file is present as:

tecmint:x:504:504:tecmint:/home/tecmint:/bin/bash

There are seven fields which are described as:

**Username:** This is the usrname that will be used to logn the system. It is 1-32 character long.

**Password:** User password (or x character) stored in /etc/shadow file in encrypted format.

**User ID (UID):** Every user must have a User ID (UID) User Identification Number. By default UID 0 is reserved for root user and UID’s ranging from 1-99 are reserved for other predefined accounts. Further UID’s ranging from 100-999 are reserved for system accounts and groups.

**Group ID (GID):** The primary Group ID (GID) Group Identification Number stored in /etc/group file.

**User Info:** This field is optional and allows you to define extra information about the user. For example, user full name. This field is filled by ‘finger’ command.

**Home Directory:** The absolute location of user’s home directory.

**Shell:** The absolute location of a user’s shell i.e. /bin/bash.

1. **Creating a user with different home directory.**

**We can use the command as:**

**useradd -d /home/VNL-HINDUSTAN tecmint**

This command will add the user tecmint in the mentioned directory.

1. **Creating a user with specific USER\_ID**

By default when we create a new user account in linux it assigns userid 500, 501 and so on.

But if we want to create a user with specific user\_id we use the command:

useradd –u 999 tecmint

1. **Creating a user with specific USER\_ID and group\_id**

useradd –u 1000 –g 500 tecmint

1. **Adding the user with account expiry date.**

If we want to add any user and set the expiry date for him/her then we can use –e option and the format of date should be like: YYYY-MM-DD

useradd –e 2015-03-25 tecmint

1. **Create user with password expiry date:**

-f option is used to define the number of days after which a password expires.

useradd –e 2015-03-25 –f 45 tecmint

**There are various other options also for creating the users but are not discussed here. If you want to learn all the options you can find at the below mentioned link:**

[**http://www.tecmint.com/add-users-in-linux/**](http://www.tecmint.com/add-users-in-linux/)

**9. How to Install and Configure SVN?**

**There are various steps that we need to follow in installing and configuring svn.**

1. First step is to install and configure the dependencies softwares.

[**https://www.youtube.com/watch?v=hC1tRzPSg5A**](https://www.youtube.com/watch?v=hC1tRzPSg5A)

**10. Questions on SQL?**

**Various questions for SQL are discussed as:**

1. **What is DBMS**

The database management system is a collection of programs that enables user to store, retrieve, update and delete information from a database.

1. **What is RDBMS**

Relational database management system (RDBMS) is a DBMS that is based on relational model. Data from relational database can be accessed or reassembled in many different ways without having to reorganize the database tables. Data from relational database can be accessed using an API , Structured Query Language (SQL).

1. **What is SQL?**

Structured Query Language (SQL) is a language designed specifically for communicating with databases.

1. **What are the types of SQL statements?**
2. **DDL** – Data Definition Language

DDL is used to define the structure that holds the data. For example, Create, Alter, Drop and Truncate table.

1. **DML**– Data Manipulation Language

DML is used for manipulation of the data itself. Typical operations are Insert, Delete, Update and retrieving the data from the table. Select statement is considered as a limited version of DML, since it can't change data in the database. But it can perform operations on data retrieved from DBMS, before the results are returned to the calling function.  
  
3. **DCL**– Data Control Language   
DCL is used to control the visibility of data like granting database access and set privileges to create tables etc. Example - Grant, Revoke access permission to the user to access data in database.

1. **What is a field in a database?**

This is an area within a record reserved for a specific piece of data.

1. **What is a record in database?**

A record is the collection of values / fields of a specific entity: i.e. an Employee, Salary etc.

1. **What is a table in a database?**

Table is a collection of records of a specific type.

1. **What is database transaction?**

A **transaction** symbolizes a unit of work performed within a [database management system](http://en.wikipedia.org/wiki/Database_management_system) (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database.

There are various properties of transaction.

1. **Atomicity**

A transaction consists of many steps. When all the steps get completed, it will get reflected in database or if any step fails, all the transactions are rolled back.

1. **Consistency**

The database will move from one consistent state to another, if the transaction succeeds and remain in the original state, if the transaction fails.

1. **Isolation**

Every transaction should operate as if it is the only transaction in the system.

1. **Durability**

Once a transaction has completed successfully, the updated rows/records must be available for all other transactions on a permanent basis.

1. **What is a database lock?**

Database lock tells a transaction that the item used in question is currently used by other transactions.

There are two types of lock in database:

1. **Shared lock**

When shared lock is applied to the data item, the other transaction can only read the item but cannot write into the data item.

1. **Exclusive lock**

When exclusive lock is applied to the data item, the other transaction cannot read and write into the data item.

1. **What is normalization in database?**

The process of removing the redundant data by splitting up the table in a well defined fashion is called as normalization.

<http://a4academics.com/interview-questions/53-database-and-sql/411-sql-interview-questions-and-answers-database?showall=&start=1>

contiue from here……………………

.**11. Windows platforms?**

**12. Questions on IIS?**

**13. How to create a branch in SVN?**

We create a branch in svn by using svn copy command.

**14. When you will create a branch?**Whenever we want to create a new line for the development and insert new features in it so as to keep the main code at the safe side then we will create a new branch.

**15. What is the structure of branches in the repo?**Branches and tags in svn are created very easily. This is done by creating a form of pointer or reference from one location to another without duplicating a lot of information.

**16. How to create backups of the repository?**

**17. Commands to create New User, New branch, and code Merging?**

**18. How you will release products to the client?**

**19. Do you know about Batch files?**In windows operating system batch file is a type of script file, a text file which contains a series of commands to be executed by the command line interpreter. There are three files extensions for batch file in windows: **.bat, .cmd, .btm**

**.bat files run with MS-DOS and every version of windows under COMMAND.COM or cmd.exe**

**.cmd runs in Windows NT family**

**.btm is used by 4DOS and 4NT. The scripts that run on 4DOS and 4NT are faster as the script is loaded entirely ready for execution rather than line by line.**

In batch file the arguments/parameters are accessed like %0, %1, %2 etc.

**20. Do you know about NSIS/Installshield?**

**21. What are the tasks you will do in daily routine?**

**22. What will do for repeated tasks?**We will make the script if we want to perform the same tasks again and again and will run the script in crontab so that task may be performed in background itself.

**23. What do you know about SVN/CVS/TFS/GIT?**

**24. How to migrate from SVN to GIT?**

**25. Questions on JOINS, HAVING, GROUP BY, Alias in SQL?**

**26. Difference between DROP, DELETE, Truncate?**DELETE command is used to delete/remove rows from a table. We can use ROLLBACK command on DELETE.

TRUNKATE is used to delete all the rows from a table. The operations performed cannot be rolled back. TRUNCATE is faster than DELETE.

DROP is used to remove a table from the database. The operations cannot be rolled back.

DROP and TRUNCATE are DDL commands whereas DELETE is a DML command.

In TRUNCATE we cannot use WHERE clause but in delete we can use the condition in WHERE clause.

**27. Difference between branch, tag, trunk?**trunk is the main line where the code is stored.

Branches are created for the development purpose. When the development is over the code is merged to the trunk and the branch is destroyed. Tag is generally a mark.

**29. ls all option.**

This command is used to list the contents in any directory. There are various options: 1. **–a option** will list the hidden files in the directory. **ls –l** option will list the details of the files and directories.

**30. Using grep display the line which don’t have “name” ( grep –v)**

**31. Different permission of file.**

The various permissions to the files are read, write and execute.

1. **File system** in SVN.

One can view the Subversion filesystem as "two-dimensional".[[10]](http://en.wikipedia.org/wiki/Apache_Subversion#cite_note-10) Two coordinates are used to unambiguously address filesystem items:

* **Path** (regular [path](http://en.wikipedia.org/wiki/Path_(computing)) of [Unix-like](http://en.wikipedia.org/wiki/Unix-like) OS filesystem)
* **Revision**

Each revision in a Subversion filesystem has its own [*root*](http://en.wikipedia.org/wiki/Root_directory), which is used to access contents at that revision. Files are stored as links to the most recent change; thus a Subversion repository is quite compact. The system consumes storage space proportional to the number of changes made, not to the number of revisions.

The Subversion filesystem uses transactions to keep changes [atomic](http://en.wikipedia.org/wiki/Atomicity_(database_systems)). A transaction operates on a specified revision of the filesystem, not necessarily the latest. The transaction has its own *root*, on which changes are made. It is then either committed and becomes the latest revision, or is aborted. The transaction is actually a long-lived filesystem object; a client does not need to commit or abort a transaction itself, rather it can also begin a transaction, exit, and then can re-open the transaction and continue using it. Multiple clients can access the same transaction and work together on an atomic change, though no existing clients expose this capability.

### Repository types[[edit](http://en.wikipedia.org/w/index.php?title=Apache_Subversion&action=edit&section=3)]

Subversion offers two types of repository storage.

1. **Berkeley DB (deprecated**[[6]](http://en.wikipedia.org/wiki/Apache_Subversion#cite_note-6)**)**[[edit](http://en.wikipedia.org/w/index.php?title=Apache_Subversion&action=edit&section=4)]
2. The original development of Subversion used the [Berkeley DB](http://en.wikipedia.org/wiki/Berkeley_DB) package. Subversion has some limitations with Berkeley DB usage when a program that accesses the database crashes or terminates forcibly. No data loss or corruption occurs, but the repository remains offline while Berkeley DB replays the journal and cleans up any outstanding locks. The safest way to use Subversion with a Berkeley DB repository involves a single server-process running as one user (instead of through a shared filesystem).[[7]](http://en.wikipedia.org/wiki/Apache_Subversion#cite_note-backend-7)
3. **FSFS**[[edit](http://en.wikipedia.org/w/index.php?title=Apache_Subversion&action=edit&section=5)]
4. In 2004, a new storage subsystem was developed and named FSFS. It works faster than the Berkeley DB backend on directories with a large number of files and takes less disk space, due to less logging.[[7]](http://en.wikipedia.org/wiki/Apache_Subversion#cite_note-backend-7)
5. Beginning with Subversion 1.2, FSFS became the default data store for new repositories.
6. The etymology of "FSFS" is based on Subversion's use of the term "filesystem" for its repository storage system. FSFS stores its contents directly within the operating system's filesystem, rather than a structured system like Berkeley DB. Thus, it is a "[Subversion] FileSystem atop the FileSystem".
7. Write ant script for create war file.
8. Own logging system in apache tomcat.
9. **migration from one version control to another( ex. svn to git).**

These files are simply zipped files using java jar tool. These files are created for different purposes. Here is the description of these files:

* **.jar files:** The .jar files **contain the libraries, resources and accessories files** like property files.
* **.war files:** The war file **contains the web application** that can be deployed on the any servlet/jsp container. The .war file **contains jsp, html, javascript** and other files for necessary for the development of web applications.

**About RPM (software packing tools)**

<http://www.rpm.org/max-rpm/ch-rpm-install.html>

<http://www.oracle-base.com/articles/linux/linux-build-simple-rpm-packages.php>

**Httpd-Apache**

<http://httpd.apache.org/docs/current/install.html> ( for linux)

**Linux-Apache authentication**

<http://www.yolinux.com/TUTORIALS/LinuxTutorialApacheAddingLoginSiteProtection.html>

**Nolio, Rational jazz, urbancode**

Ant debug

Behavior driven programming

Test driven programming

Diff between ci build and night build

How build.xml call another build.xml

How to handle 30 checkin gap of 1 min in jenkin, Gradle

Remove all files between certain date.

You can also use the find command. For instance if you wanted to delete all jpg's in the current directory older then 7 days:  
  
rm -f `find \*.jpg -type f -mtime +7`  
  
If you want to test this out, you can just do this.  
  
  
echo `find \*.jpg -type f -mtime +7`  
  
This way you will know exactly what will be deleted before you run the command.  
  
Also type man find at the command line to get more information on the find command.

find **/path/to/specific/directory** -type f -mtime **+30** -maxdepth 1 -exec rm {} \;

How to sync local server to remote server.

How to resolve compilation dependencies on jar file.

How to solve circular dependencies in ant script.

Redirected tool in apache

Web server vs application server

How to know which application run on given port no in linux

Database import,export ,replication

How to set cluster in apach-tomcat

**Reverse string in linux:** rev

**See environment variable in linux :** printenv

How to convert http to https in apache-tomct ?

How to use SSL ?

1. Diff between locate and find ?

**Ans:** Locate and find both commands will find file. But each of them work in a quite different way.

'locate' will work in a off line mode: For a simple explanation, a file indexing database in Unix system called 'slocate'. This database will have all your file location lists which ships with Unix system. When you execute locate then it will go to that database and search for a particular file.

Now the problem with locate command is that if you just create a file/files form where you want to search a file then locate will not work because the slocate database is not updated yet.

In order to overcome this problem you can use updatedb which will eventually update the slocate database. Now executing the command again will find the newly created file/files that you just have created before.

That's why many Linux system administrator use the corn job to update the slocate database on a regular basis.

'find' will work in a on line/ real time mode. It will actually go and search all the directories to find the particular file specified and it examine each file one by one. Therefore, it requires a lot of I/O calls.

So based on the nature, it is clear that locate is faster than find but find is real time.

1. Parse file line by line in vi editor (it should display in proper manner)?
2. How to retrain history of repo (when redeploy or server crash) ?
3. How to setup Apache ?
4. What happened in background when and command like ‘ls’ run in linux ?

**Ans:**

When you execute a program on your UNIX system, the system creates a special environment for that program. This environment contains everything needed for the system to run the program as if no other program were running on the system.

Whenever you issue a command in UNIX, it creates, or starts, a new process. When you tried out the ls command to list directory contents, you started a process. A process, in simple terms, is an instance of a running program.

The operating system tracks processes through a five digit ID number known as the pid or process ID . Each process in the system has a unique pid.

Pids eventually repeat because all the possible numbers are used up and the next pid rolls or starts over. At any one time, no two processes with the same pid exist in the system because it is the pid that UNIX uses to track each process.

1. Diff between daemon and fork ?

**Ans:**

Daemons are system-related background processes that often run with the permissions of root and services requests from other processes.

A daemon process has no controlling terminal. It cannot open /dev/tty. If you do a "ps -ef" and look at the tty field, all daemons will have a ? for the tty.

More clearly, a daemon is just a process that runs in the background, usually waiting for something to happen that it is capable of working with, like a printer daemon is waiting for print commands.

If you have a program which needs to do long processing then its worth to make it a daemon and run it in background.

1. How to connect linux to window and vise-versa ?

**Ans:** using cygwin or gnuwin32

**From linux to window**

sshpass -p 'vnlscm123' ssh cmadmin@10.100.208.110 << EOF

cmd

cd C:\EDS\_FRMWK

**From window to linux**

ssh [cmbuild@10.100.211.61](mailto:cmbuild@10.100.211.61) and scp from batch script.

1. Run SQL script for getting all data through command line ?

**Ans:**

mysql -u root -proot123 cmis -e "select \* from vvdn;" ( **for same machine**).

mysql -h 10.100.211.63 -u root -proot123 cmis -e "select \* from vvdn;" ( **for different machine**)

1. How to open large file ( 3 gb) in linux ?

**Ans:**

## About split

Split a [file](http://www.computerhope.com/jargon/f/file.htm) into pieces.

## Syntax

split [*OPTION*]... [*INPUT* [*PREFIX*]]

## Description

**split** outputs fixed-size pieces of input *INPUT* to files named *PREFIX***aa**, *PREFIX***ab**, ...  
  
The default size for each split file is 1000 lines, and default PREFIX is "**x**". With no INPUT, or when INPUT is a dash ("**-**"), read from [standard input](http://www.computerhope.com/jargon/s/stdin.htm).

## Options

|  |  |
| --- | --- |
| **-a** *N*, **--suffix-length=***N* | Use [suffixes](http://www.computerhope.com/jargon/s/suffix.htm) of length *N* (default **2**) |
| **-b** *SIZE*, **--bytes=***SIZE* | Write *SIZE* [bytes](http://www.computerhope.com/jargon/b/byte.htm) per output file. |
| **-C** *SIZE*, **--line-bytes=***SIZE* | Write at most *SIZE* bytes of lines per output file. |
| **-d**, **--numeric-suffixes** | Use numeric suffixes instead of alphabetic. |
| **-e**, **--elide-empty-files** | Do not generate empty output files with "**-n**" |
| **--filter=***COMMAND* | Write to [shell](http://www.computerhope.com/jargon/s/shell.htm) command *COMMAND*; file name is **$FILE** |
| **-l** *NUMBER*, **--lines=***NUMBER* | Put *NUMBER* lines per output file. |
| **-n** *CHUNKS*, **--number=***CHUNKS* | Generate *CHUNKS* output files. (See below.) |
| **-u**, **--unbuffered** | Immediately copy input to output with "**-n r/**...". |
| **--verbose** | Print a [verbose](http://www.computerhope.com/jargon/v/verbose.htm) diagnostic just before each output file is opened. |
| **--help** | Display a help message and exit. |
| **--version** | Output version information and exit. |

*SIZE* may be one of the following, or an [integer](http://www.computerhope.com/jargon/i/integer.htm) optionally followed by one of following multipliers:

|  |  |
| --- | --- |
| **Suffix** | **multiplier** |
| **KB** | **1000** |
| **K** | **1024** |
| **MB** | **1000** x **1000** |
| **M** | **1024** x **1024** |

...and so on for **G** ([gigabytes](http://www.computerhope.com/jargon/g/gb.htm)), **T** ([terabytes](http://www.computerhope.com/jargon/t/tb.htm)), **P** ([petabytes](http://www.computerhope.com/jargon/p/petabyte.htm)), **E** ([exabytes](http://www.computerhope.com/jargon/e/eb.htm)), **Z**([zettabytes](http://www.computerhope.com/jargon/z/zettabyt.htm)), **Y** ([yottabytes](http://www.computerhope.com/jargon/y/yottabyt.htm)).

*CHUNKS* may be:

* *N*: split into *N* files based on size of input
* *K***/***N*: output *K*th of *N* to standard output
* *l***/***N*: split into *N* files without splitting lines
* *l***/***K***/***N*: output *K*th of *N* to standard output without splitting lines
* *r***/***N*: like "**l**" but use round robin distribution *r***/***K***/***N* likewise but only output *K*th of*N* to standard output

## Examples

## About split

Split a [file](http://www.computerhope.com/jargon/f/file.htm) into pieces.

## Syntax

split [*OPTION*]... [*INPUT* [*PREFIX*]]

## Description

**split** outputs fixed-size pieces of input *INPUT* to files named *PREFIX***aa**, *PREFIX***ab**, ...  
  
The default size for each split file is 1000 lines, and default PREFIX is "**x**". With no INPUT, or when INPUT is a dash ("**-**"), read from [standard input](http://www.computerhope.com/jargon/s/stdin.htm).

*SIZE* may be one of the following, or an [integer](http://www.computerhope.com/jargon/i/integer.htm) optionally followed by one of following multipliers:

|  |  |
| --- | --- |
| **Suffix** | **multiplier** |
| **KB** | **1000** |
| **K** | **1024** |
| **MB** | **1000** x **1000** |
| **M** | **1024** x **1024** |

...and so on for **G** ([gigabytes](http://www.computerhope.com/jargon/g/gb.htm)), **T** ([terabytes](http://www.computerhope.com/jargon/t/tb.htm)), **P** ([petabytes](http://www.computerhope.com/jargon/p/petabyte.htm)), **E** ([exabytes](http://www.computerhope.com/jargon/e/eb.htm)), **Z**([zettabytes](http://www.computerhope.com/jargon/z/zettabyt.htm)), **Y** ([yottabytes](http://www.computerhope.com/jargon/y/yottabyt.htm)).

*CHUNKS* may be:

* *N*: split into *N* files based on size of input
* *K***/***N*: output *K*th of *N* to standard output
* *l***/***N*: split into *N* files without splitting lines
* *l***/***K***/***N*: output *K*th of *N* to standard output without splitting lines
* *r***/***N*: like "**l**" but use round robin distribution *r***/***K***/***N* likewise but only output *K*th of*N* to standard output

## Examples

split -b 22 newfile.txt new

Split the file **newfile.txt** into three separate files called **newaa**, **newab** and**newac**..., with each file containing 22 bytes of data.

split -l 300 file.txt new

Split the file **newfile.txt** into files beginning with the name **new**, each containing 300 lines of text.

Split the file **newfile.txt** into files beginning with the name **new**, each containing 300 lines of text.

1. What exactly in background when open any website ?

**Ans:**

* Browser splits what you type (the URL) into a hostname and a path.
* Browser forms an HTTP request to ask for the data at the given hostname and path.
* Browser performs DNS lookup to resolve the hostname into an IP address.
* Browser forms a TCP/IP connection to the computer specified via the IP address. (This connection is actually formed out of many computers, each passing the data along to the next.)
* Browser sends the HTTP request down the connection to the given IP address.
* That computer receives the HTTP request from the TCP/IP connection and passes it to the web server program.
* Web server reads the hostname and path and finds or generates the data that you've asked for.
* Web server generates an HTTP response containing that data.
* Web server sends that HTTP response back down the TCP/IP connection to your machine.
* Browser receives the HTTP response and splits it into headers (describing the data) and the body (the data itself).
* Browser interprets the data to decide how to display it in the browser - typically this is HTML data that specifies types of information and their general form.
* Some of the data will be metadata that specifies further resources that need to be loaded, such as style sheets for detailed layout, or inline images, or Flash movies. This metadata is specified again as a URL, and this whole process repeats for each one until all are loaded.

1. Crontab run with 30 second gap ?

**Ans:** Use two same command of crontab within sleep 30 in shell script.

1. Soft link vs hardlink

**Ans:** Link are of two types Soft links (symbolic links ) or Hard link

1. **Soft Links** (symbolic links )

You can make links for files & folder & you can create link (shortcut) on different partition & got different inode number from original.

If *real* copy is deleted the link **will not work**.

1. **Hard Links**

For files only & you cannot create on different partition ( it should be on same partition ) & got same inode number as original

If the *real* copy is deleted the **link will work** (because it act as original file )

**Question:** How to make soft link ?

**Answer:** A soft link can be made with ln -s, 1st you need to define the source and then you need to define the destination ( keep it mind you need to define the full path of source and destination otherwise it will not work )

sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib32/libGL.so.1

(----------Source-------) ( Destination )

enter image description here

As you can see if has different inode and can be made on different partition

**Question:** How to make Hard link ?

**Answer:** A Hard link can be made with ln , 1st you need to define the source and then you need to define the destination ( keep it mind you need to define the full path of source and destination otherwise it will not work )

I have a script in /script folder name firefox

ls -i ( Shows you the inode )

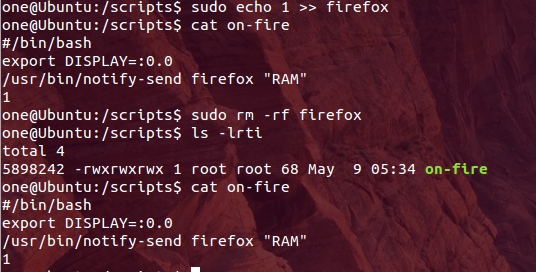
5898242 firefox

ln /scripts/firefox /scripts/on-fire

( Source ) ( Destination )

enter image description here

As you can see it has same inode and if I delete the original one the link will work, as it act as original



1st i have check that link is working or not, then i have deleted firefox script

**You Question :** It would be nice if someone could provide a setting where hard link might be preferable over a symbolic link.

**Answer** : Depending on disk partition layout is, **Hard Link** have limitation must be on same partition **( -1 point )** and can be of file **( -1 point )** ) but **+1 point** is that if original is deleted the link will work as it act as original

whereas soft link can be made of folders & files (+1 point) , No partition limitation (+1 point), But only**(-1 point)** is that if source is deleted the link will not work

1. How sshd works
2. How to know which port running which service

**Ans:** netstat -a | grep 80

1. What is zombie process

**Ans: What is a zombie process?**

When a process finishes execution, it will have an exit status to report to its parent process. Because of this last little bit of information, the process will remain in the operating system’s process table as a zombie process, indicating that it is not to be scheduled for further execution, but that it cannot be completely removed (and its process ID cannot be reused) until it has been determined that the exit status is no longer needed.

When a child exits, the parent process will receive a SIGCHLD signal to indicate that one of its children has finished executing; the parent process will typically call the wait() system call at this point. That call will provide the parent with the child’s exit status, and will cause the child to be reaped, or removed from the process table.

**How do I see if there are zombie processes on a system?**

Run “ps aux” and look for a Z in the STAT column.

**How do I remove zombie processes from a system?**

Well, first you can wait. It’s possible that the parent process is intentionally leaving the process in a zombie state to ensure that future children that it may create will not receive the same pid. Or perhaps the parent is occupied, and will reap the child process momentarily.

Secondly, you can send a SIGCHLD signal to the parent (“kill -s SIGCHLD <ppid>“). This will cause well-behaving parents to reap their zombie children.

Finally, you can kill the parent process of the zombie. At that point, all of the parent’s children will be adopted by the init process (pid 1), which periodically runs wait() to reap any zombie children.