**Class 9**

A repository is essentially a server that acts as a library/storage for tools and updates that can be utilized by CentOS. When you use yum to install something such as Apache, you are connecting to a repository server – this could be a local private server that a company is using or a public one on the web.

You have two ways to add more Centos **repositories**:

* Drop .repo files into your /etc/yum.repos.d/ directory
* Add a repository entry in your /etc/yum.conf file

<http://www.cyberciti.biz/faq/centos-fedora-redhat-yum-repolist-command-tolist-package-repositories/>

<http://www.cyberciti.biz/tips/redhat-centos-fedora-linux-setup-repo.html>

<http://www.cyberciti.biz/tips/rhel5-fedora-core-add-new-yum-repository.html>

https://wiki.centos.org/AdditionalResources/Repositories

**GRUB** & **LILO** bootloaders – popular tools used for multi-booting operating systems.

**Chkconfig** (to disable a service permanently or to automatically start it when the server reboots)

<http://www.cyberciti.biz/faq/check-running-services-in-rhel-redhat-fedora-centoslinux/>

example use:

chkconfig httpd on (this enables the Apache service when the server boots up)

**Linux Runlevels Explained** (source: http://www.liquidweb.com/kb/linux-runlevels-explained/)

A **runlevel** is one of the modes that a Unix -based operating system will run in. Each runlevel has a certain number of services stopped or started, giving the user control over the behavior of the machine. Conventionally, seven runlevels exist, numbered from zero to six.

After the Linux kernel has booted, the init program reads the **/etc/inittab** file to determine the behavior for each runlevel. Unless the user specifies another value as a kernel boot parameter, the system will attempt to enter (start) the default runlevel.

|  |  |  |
| --- | --- | --- |
| Run Level | Mode | Action |
| 0 | Halt | Shuts down system |
| 1 | Single-User Mode | Does not configure network interfaces, start daemons, or allow non-root logins |
| 2 | Multi-User Mode | Does not configure network interfaces or start daemons. |
| 3 | Multi-User Mode with Networking | Starts the system normally. |
| 4 | Undefined | Not used/User-definable |
| 5 | X11 | As runlevel 3 + display manager(X) |
| 6 | Reboot | Reboots the system |
| Standard run levels for Red Hat based distributions | | |

Most Linux servers lack a graphical user interface and therefore start in runlevel 3. Servers with a GUI and desktop Unix systems start runlevel 5. When a server is issued a reboot command, it enters runlevel 6.

Init scripts

**Init** (short for initialization) is the program on Unix and Unix-like systems that spawns all other processes. It runs as a daemon and typically has PID 1.

The /etc/inittab file is used to set the default run level for the system. This is the runlevel that a system will start up on upon reboot. The applications that are started by init are located in the **/etc/rc.d** folder. Within this directory there is a separate folder for each run level, eg *rc0.d*, *rc1.d*, and so on.

chkconfig

The **chkconfig** tool is used in Red Hat based systems (like CentOS) to control what services are started at which runlevels. Running the command *chkconfig –list* will display a list of services whether they are enabled or disabled for each runlevel.

root@host:~ # chkconfig --list  
filelimits 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
syslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
gpm 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
cpanel 0:off 1:off 2:off 3:on 4:on 5:on 6:off  
kudzu 0:off 1:off 2:off 3:on 4:on 5:on 6:off  
ntpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off  
netfs 0:off 1:off 2:off 3:on 4:on 5:on 6:off  
network 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
netplugd 0:off 1:off 2:off 3:off 4:off 5:off 6:off  
rawdevices 0:off 1:off 2:off 3:on 4:on 5:on 6:off  
ipchains 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
anacron 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
cups 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
xfs 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
xinetd 0:off 1:off 2:off 3:on 4:on 5:on 6:off  
httpd 0:off 1:off 2:off 3:on 4:off 5:on 6:off

**Single User mode** is a mode that a multi-user system (like a Linux server) can be booted into the operating system as a superuser. Booting a system into this mode does not start networking, but can be used to make changes to any configuration files on the server. One of the most common usages for single-user mode is to change the root password for a server on which the current password is unknown.

Runlevels are an important part of the core of the Linux operating system. While not something the average administrator will work with on a daily basis, understanding runlevels gives the administrator another layer of control and flexibility over the servers they manage.

**Fsck** (file system check) – same as Windows’ version of drive check for errors. You’d want to use this in “single user mode.”

**RPM**: a package management system (if yum doesn’t work, you can use this): <http://www.cyberciti.biz/faq/rhel-redhat-fedora-opensuse-linux-install-rpmfile-command/>

Example scenario when this could come in handy: no Internet access to your server and you need to install something or fix yum. You would manually add an rpm file to the server (either physically or via a shared drive).

Good **yum** commands resource: <http://daddy-linux.blogspot.com/2012/02/yum-command-update-install-packages.html>

**sed :** *Sed* is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline). While in some ways similar to an editor which permits scripted edits (such as *ed*), *sed* works by making only one pass over the input(s), and is consequently more efficient. But it is *sed*'s ability to filter text in a pipeline which particularly distinguishes it from other types of editors.

https://www.digitalocean.com/community/tutorials/the-basics-of-using-the-sed-stream-editor-to-manipulate-text-in-linux

<http://en.wikipedia.org/wiki/Sed>

<http://linux.about.com/od/commands/l/blcmdl1_sed.htm>

**AWK**:

awk is a complete pattern scanning and processing language, it is most commonly used as a Unix command-line filter to reformat the output of other commands.

<http://www.thegeekstuff.com/2010/01/awk-introduction-tutorial-7-awk-print-examples/>

http://www.tutorialspoint.com/awk/

https://www.gnu.org/software/gawk/manual/html\_node/Very-Simple.html

<http://www.cyberciti.biz/faq/tag/awk-command/>

**FTP** – File Transfer Protocol - used to transfer files, most commonly to update websites or share files with others. (**vsftpd** are two you could install).

**Apache** – the defacto Linux web server (displays the web pages that we surf). Stats on usage: http://w3techs.com/technologies/details/ws-apache/all/all

**Nginx** – another web server gaining popularity that is UNIX based.

**SMTP** – Simple Mail Transfer Protocol – de facto standard for mail transport across the Internet.

**POP** – Post Office Protocol for incoming email.

Email things to look into: SPF records, PTR records (DNS records), MX records, DKIM.

**IMAP** – Gmail, Yahoo, Hotmail etc. have that – so that you can view your emails anywhere. POP is meant to download emails to one machine.

**SSL certificates** – those encrypt websites, the little lock sign you see or “Verified by Verisign” messages. Geotrust is another big company that sells them.

**NFS** – Network File System is the UNIX/Linux way of sharing files and applications across the network. You could share a drive using NFS and multiple servers could access it (mount it).

**NIS** – Network Information System – facilitates the sharing of critical data stored in flat files among systems on a network. Typically, files such as /etc/passwd and /etc/group, which ideally would remain uniform across all hosts, are shared via NIS.

**Samba** – allows UNIX-based systems to interoperate with Windows-based and other operating systems. Provides file & print sharing services to Windows clients. Uses the SMB/CIFS protocol suite.

**LDAP** – Lightweight Directory Access Protocol (same as Active Directory but Linux version). OpenLDAP is free.

**Printing** – Relates to Common UNIX Printing System (CUPS).

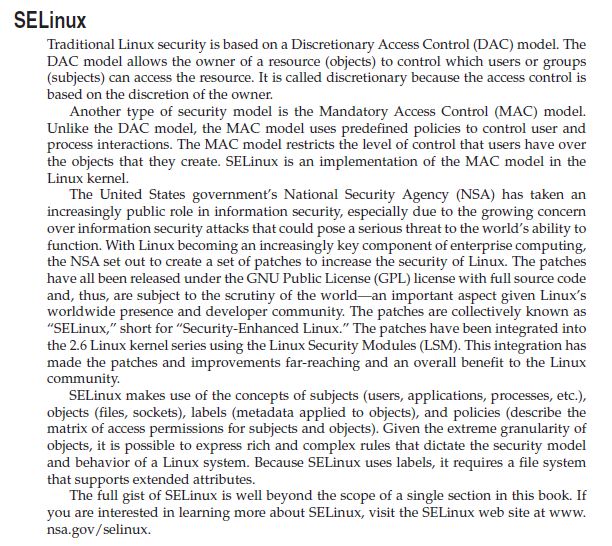
**DHCP** – Dynamic Host Configuration Protocol – this system assigns IPs automatically rather than having to manually add them on each machine.

**NAS** – Network-attached storage – basically a shared drive over the network (versus being a DAS – direct-attached storage). It also provides a file system.

**SAN** – Storage Area Network – provides block-based storage and leaves file system concerns on the “client” side.

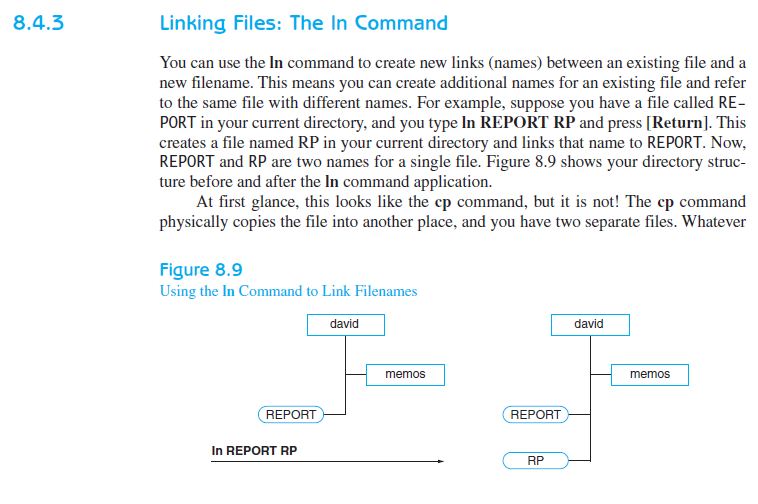
Despite their differences, SAN and NAS are not mutually exclusive, and may be combined as a SAN-NAS hybrid, offering both file-level protocols (NAS) and block-level protocols (SAN) from the same system. An example of this is Openfiler, a free software product running on Linux-based systems. A shared disk file system can also be run on top of a SAN to provide filesystem services.

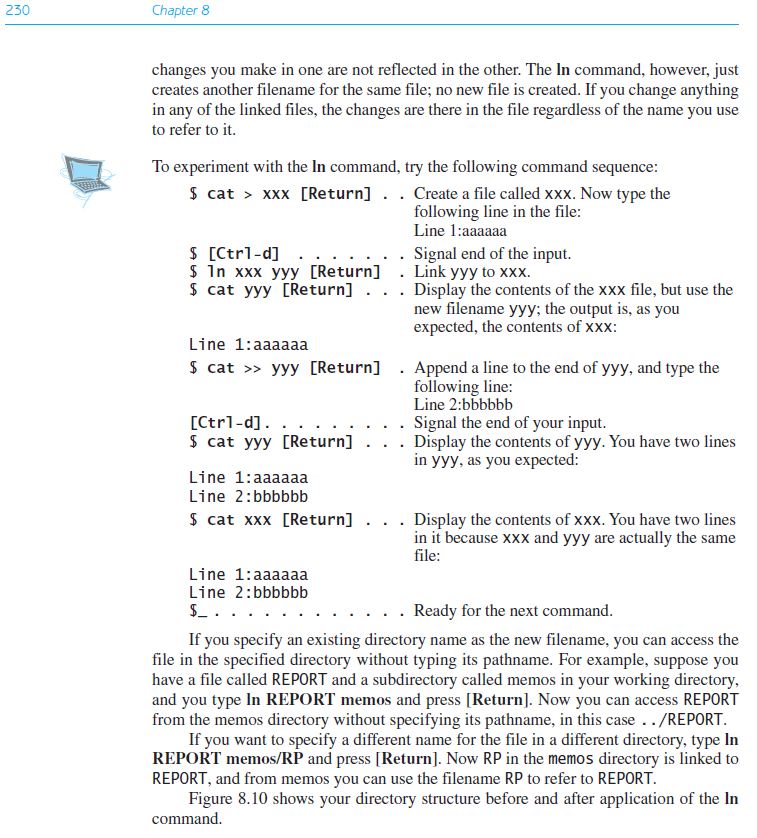
**Selinux:**



http://en.wikipedia.org/wiki/Security-Enhanced\_Linux

**Linking files**





Good URL describing the differences between a symbolic link (soft) and hard links:

<http://www.cyberciti.biz/tips/understanding-unixlinux-symbolic-soft-and-hard-links.html>

