

# SCL System Administration

Part 1 (update 2)



# 1. Linux History

(1991-2015)



# Linux History 1

Year	Event
1991	Linus Torvalds announces Linux Kernel 8/21/91
1992	Linux Kernel Relicensed under GNU/GPL
1993	Slackware, Debian Distributions Released
1994	Linux Kernel V1 Released; Xfree86 offers a GUI; Redhat Linux V1.0, <b>SUSE Linux V1.0</b> both released
1996	Linux Kernel V2 Released with Symmetric Multi-Processing (SMP)
1998	KDE GUI Development begins (via Qt toolkit)
1999	Gnome Development begins (KDE replacement)
2003	Novell buys SuSE Company, brands, trademarks
2005	OpenSUSE project and free distribution begins;



# Linux History 2

## Year            Event

2006	Novell and Microsoft Announce Cooperation agreement for interoperability, patent protection
2011	Attachmate Group acquires Novell (and SUSE); Linux Kernel V3 released
2015	Linux Kernel V4 released

## References

**SUSE [Software und System-Entwicklung or Software and System Development ] History**

<<https://www.suse.com/company/history/>>

**SUSE Linux Distributions**

<[https://en.wikipedia.org/wiki/SUSE Linux distributions](https://en.wikipedia.org/wiki/SUSE_Linux_distributions)>



# SUSE Linux Releases

<b>Name</b>	<b>Version</b>	<b>Date</b>	<b>End of Life</b>	<b>Kernel</b>
<b>SLES</b>	<b>11</b>	<b>3/09</b>		<b>2.6.27</b>
	<b>11SP3</b>	<b>7/13</b>		<b>3.0.76</b>
	<b>12</b>	<b>10/14</b>		<b>3.12.28</b>
<b>open SUSE</b>	<b>11.4</b>	<b>3/11</b>	<b>July 2015</b>	<b>2.6.37</b>
	<b>12.3</b>	<b>3/13</b>	<b>Jan. 2015</b>	<b>3.7.10</b>
	<b>13.1</b>	<b>11/13</b>	<b>May 2015</b>	<b>3.11.6</b>
	<b>13.2</b>	<b>11/14</b>	<b>Nov. 2016</b>	<b>3.16.6</b>



# Releases References

- ◆ **SLES:** <[https://en.wikipedia.org/wiki/SUSE\\_Linux\\_Enterprise\\_Server#Version\\_history](https://en.wikipedia.org/wiki/SUSE_Linux_Enterprise_Server#Version_history)>
- ◆ **SUSE Linux Distributions:**  
<[https://en.wikipedia.org/wiki/SUSE\\_Linux\\_distributions](https://en.wikipedia.org/wiki/SUSE_Linux_distributions)>
- ◆ **SLES Kernel Versions**  
<[https://wiki.novell.com/index.php/Kernel\\_versions](https://wiki.novell.com/index.php/Kernel_versions)>
- ◆ **SLES Technical Information (Kernel Limits)**  
<<https://www.suse.com/products/server/technical-information/#Kernel>>
- ◆ **SLES 12 Release Notes**  
<[https://www.suse.com/releasenotes/x86\\_64/SUSE-SLES/12/](https://www.suse.com/releasenotes/x86_64/SUSE-SLES/12/)>



## 2. Linux Command Help Aids

man, info, help, HOWTOs  
*command (unix)* [site:wikipedia.org](http://site:wikipedia.org)



# **man** (manual pages)

- ◆ A program that uses a files database stored locally, on a network or the web, to retrieve a write-up about one Linux command or topic. Access via:
- ◆ **\$ man [section No.] <commandname>**
- ◆ **\$ man man; man intro; man info**



# man Command (2)

- ◆ These commands offer meta-info.
- ◆ **Search List**  
\$ man -k “keyword” or  
\$ apropos “keyword”
- ◆ **One Line Description**  
\$ man -f “command”  
\$ whatis “command”



# info and help

- ◆ **info** An online manual reader for GNU Software (Linux)

**\$ info “linux command”**

**\$ info info; info -k “keyword”**

- ◆ **help** Shows a summary for **bash** shell builtin commands

**\$ help [“bash builtin”] ; help help**



# Linux HOWTOs

- ◆ Relatively dated (1997-2010) but pertinent written explanations of Linux characteristics, features, processes and procedures.
- ◆ Writeups by experts + reader contributions occur
- ◆ Example: Bash Prompt HOWTO. See [www.linuxhowtos.org/System/Bash%20Prompt%20Howto.htm](http://www.linuxhowtos.org/System/Bash%20Prompt%20Howto.htm)
- ◆ References: HOWTO Single List: <http://www.tldp.org/HOWTO/HOWTO-INDEX/howtos.html> HOWTOs Home: [linuxhowtos.org/](http://linuxhowtos.org/)



# Web Search

- ◆ Launch a browser window and search using a search engine by typing:

***“command (unix)”* site: en.wikipedia.org**

- ◆ A Separate window for showing usage frees your shell window to enter the proper commandline you want.



# 3. YaST Facility

yast yast2 autoyast



# YaST Manage Users u1

- ◆ Commandline: `$ sudo /usr/sbin/yast2 users`  
**[shift]tab**-traverse selections **[reverse]**  
**arrow keys**-move to subselections; **<alt>Q** -Quit  
**space/enter**-choose item ; **<alt>YellowLetter**-Jump to item  
Bold white border is active Frame
- ◆ Tabs: [Users] [Groups] [Defaults] [Authentication]  
Options for these:
  - User Data** - Create Userid+password
  - Details** - i.e. Home directory path, Groups
  - Password setting/aging** (Use **pwgen**; **LDAP** > **NIS**)
  - Plug-ins** - i.e. Storage, Inode Quotas for Users (**repquota**)
- ◆ Can Create, Modify or Delete Users
- ◆ See: [doc.opensuse.org/documentation/html/openSUSE\\_114/opensuse-reference/cha.y2.userman.html](http://doc.opensuse.org/documentation/html/openSUSE_114/opensuse-reference/cha.y2.userman.html)



# YaST Commandline

- ◆ `yast -h` # Get list of command options
- ◆ `yast -l` # Get list of module names
- ◆ `yast <module>` # start module named
- ◆ `yast -i <package-name>` # install package
- ◆ `yast <module> help` # Get module help
- ◆ Some modules don't offer command line support



# Observation

**Linux: the OS with a CLUE...  
Command Line User  
Environment**



# YaST Manage Software

- ◆ Installs Software as packages that are in repositories known based on a repository index service (RIS).
- ◆ Search by name, description, RPM/package group, and pattern (= set of packages offering some functionality).
- ◆ Software categories:
  - Online Update** [YOU]
  - Software Management** [of packages] Filter Options:
    - o Patterns o Languages o RPM Groups o Repositories o Search
    - o Installation Summary o Package Classification
  - Add-on Products** [i.e. High Availability Extension]
  - Media Check** [verify error-free media]
  - Software Repositories** [Define from installation media (iso images), update repositories (RPM Files) ] [Use autorefresh, priority (conflict resolution), Keep Downloaded Packages options]
- ◆ Use **SUSE Manager** [deploys, manages, monitors SW] when many servers have to be managed automatically



# YaST Manage Software 2

- ◆ In the **Action menu** you can select **Locked** packages (never to be upgraded); **Taboo** packages (never to be installed)
- ◆ In the **Dependencies menu**, **Automatic Dependency Check** is the default. You can also select: **System Verification Mode** to assure compatibility between current system and selected packages.
- ◆ In the **View menu**, you can see the package installation information, **File List** to see the exact contents of the package, **Dependencies List** to see preview of requirements before package can be installed.
- ◆ After package selection for installation,
  - A **Dependencies Overview** is shown, with problem and possible solutions for you to choose. 1st one is best one usually.
  - If no problems, a list is shown of automatic changes that will be applied.
  - Click OK to install



# YaST Manage Devices U2

- ◆ The modular Kernel (runs outside of Linux) interacts directly with the Server's Hardware and Devices.
- ◆ `/sbin/modinfo (8)` shows information about Linux Kernel modules (See `/sbin/lsmmod (8)` for status & a list )
- ◆ Hardware Device Drivers and Users communicate via logical device files, all found in `/dev`
- ◆ `{processes, apps, commands}<---system calls via glibc---`  
`--->Kernel<--->Hardware`
- ◆ `/bin/lsblk (8)` Lists block devices in tree format  
`/sbin/hdparm (8)` Get/Set SATA/IDE device parameters.  
Beware, there are **very** dangerous options within.  
`$ /sbin/hdparm -i /dev/sda # get information about /dev/sda`



# sysfs, /sys for Devices u2

- ◆ **sysfs** is a virtual, in memory file system describing available devices, configurations and their state.
- ◆ **/sys** is the mount point of **sysfs**
- ◆ **/proc** also holds device-specific information  
(`$ cat /proc/devices | less`)



# /sys Subdirectories

Directory (subsystem)	Description
<b>block</b>	block oriented devices like Hard Disks
<b>bus</b>	Buses known to the kernel { PCI-E, SCSI, USB }
<b>class</b>	A functional tree organization: e.g. sound, graphic cards, input devices, network interfaces
<b>dev</b>	Device information for character and block devices
<b>devices</b>	all discovered devices
<b>firmware</b>	Interfaces to platform-specific subsystems {ACPI}
<b>fs</b>	A directory for a selected subset of filesystems that the kernel knows
<b>kernel</b>	Kernel Internals (e.g. cache, virtual memory status)
<b>module</b>	Dynamic modules loaded by the Kernel
<b>power</b>	Details about the system's power state [unused]



# udevadm Command

- ◆ **udevadm (8)** queries device information, triggers events, controls the udevd daemon, monitors udev and kernel events.
- ◆ Purpose is to build and test (udev) rules
- ◆ Form: **udevadm info | trigger | settle | control | hwdb | monitor | test | test-builtin # used most by SAs;**
- ◆ **control** option starts/stops **udev** daemon (reloads all rules named **nn.\*.rules** in **/etc/udev/rules.d**)  
**monitor** option displays events as they occur
- ◆ Use when new devices arise to be mounted and operate on Servers.



# udev process U2

- ◆ `/sbin/udev` -- daemon (SLES 11.x) [`systemd-udev` (SLES 12.x)] is a daemon that dynamically manages kernel detection and initialization (updating `/dev`) and disconnection of devices (removal)
- ◆ `$ /sbin/udevadm monitor [--env]` shows views of the events, timings, [environment] by `udev` process
- ◆ Kernel events to `udev` are handled by rules. See the prioritized rules set in the directory `/etc/udev/rules.d`
- ◆ `udev` sends messages to syslog. The priority level managed in `/etc/udev/udev.conf`
- ◆ Reference: [doc.opensuse.org/documentation/html/openSUSE\\_114/opensuse-reference/cha.udev.html#sec.udev.devdir](http://doc.opensuse.org/documentation/html/openSUSE_114/opensuse-reference/cha.udev.html#sec.udev.devdir)



# YaST Manage Hardware u1

- ◆ Hardware connected to the Server may be configured beginning at OS installation time and thereafter.
- ◆ Probe all hardware information via:  
`$ sudo /usr/sbin/yast2 hwinfo # save to file encouraged`
- ◆ Other Hardware related modules include:  
**disk, hwinfo, keyboard, lan, printer, scanner**
- ◆ Terminal Commands:  
**lspci (8)** - List all PCI Devices, Buses [ <?> for missing info ]  
**lsusb (8)** - List all USB Devices  
**usb-devices (1)** - Lists USB buses and device connection details  
**lscpu (8)** - List all CPUs [overview based on 1st CPU;  
vm hardware can be inaccurate]  
**hwinfo (8)** - Probe for Hardware. Preserve in a log via:  
`$ hwinfo --all --log hwinfo-out.log`



# Observation

- ◆ Being a Linux user is sort of like living in a house inhabited by a large family of carpenters and architects. Every morning when you wake up, the house is a little different. Maybe there's a new window or some walls have moved. Or perhaps someone has temporarily removed the floor under your bed. ~~ Rob Riggs



# 4. Account Management

gnome-terminal, xterm,  
useradd usermod userdel yast2 user  
passwd /etc/passwd /etc/group



# Gnome GUI to Terminal

Gnome	Terminal	Result
Open Run Command Window<Alt>F2 type: gnome-terminal	New:<Shift><Ctrl>N Exit:<Shift><Ctrl>W All: <Shift><Ctrl>Q	pty terminal window
Open Run Command Window<Alt>F2 type: xterm	\$ exit	xterm (:0) display window
Open Terminal Screen: <Ctrl><Alt>Fi {i=1-4}	Exit: <Ctrl><Alt>F7	tty terminal logical screen i
Open Console Screen: <Ctrl><Alt>F10	Exit: <Ctrl><Alt>F7	Read only console screen



# useradd(8) Command U2

- ◆ Adding new users:
  - Update `/etc/passwd` and `/etc/shadow`
  - Append User's group id to `/etc/group`
  - Set an initial password
  - Create new `/home` directory with user as owner and proper permissions
  - Copy default startup files from `/etc/skel/` to `/home/<userid>/`
  - Set up user's mail home and aliases
  - Verify that new account works  
(login as new user, run: `pwd; ls -la` )
  - Add information to your user contact database



# Supporting Commands + Configuration files

- ◆ `/usr/sbin/useradd.local` # for local customizations
- ◆ **usermod (8)** Modify existing user information in `passwd`, `shadow` and `group` files
- ◆ **userdel (8)** Delete User Account
- ◆ **passwd (1)** Update password for current/specified user
- ◆ **chage (1)** Update password expiration date
- ◆ **groupadd (8)** Add a new group name entry to `/etc/group`
- ◆ `/etc/passwd` {edit with **vipw**}, `/etc/shadow`,  
`/etc/group` {edit with **vigr**}
- ◆ `/etc/default/useradd`, `/etc/login.defs`



# /etc/passwd Format

- ◆ `$ echo " f1 f2 f3 f4\tf5\tf6\tf7"; tail -4 /etc/passwd`  
**f1 f2 f3 f4 f5 f6 f7**  
katz:x:1000:100:Robert Katz:/home/katz:/bin/bash  
robert:x:1001:100:Robert Katz:/home/robert:/bin/bash  
katz2:x:1002:100:Robert Katz 2:/home/katz2:/bin/bash  
katz3:x:1003:100:Robert Katz 3:/home/katz3:/bin/bash
- ◆ **f1** Unique Username  
**f2** placeholder for encrypted password (now in shadow file)  
**f3** Unique Userid (UID) Number  
**f4** Primary Group id (GID) Number  
**f5** Comment Field for real user name and contact  
**f6** Initial current directory after successful login  
**f7** Program to launch after successful login (usually a shell)
- ◆ Reference: **man 5 passwd**



# /etc/shadow Format u2

- ◆ `$ echo " f1 f2\t\t\tf3\tf4 f5 f6 f7 f8 f9"; sudo tail -4 /etc/shadow`  
**f1 f2** **f3 f4 f5 f6 f7 f8**  
katz:\$2y\$2charSalt\$53charEncryptedpassword:16547:0:99999:7::  
robert: \$2y\$2charSalt\$53charEncryptedpassword:16548:0:99999:7::  
katz2: \$2y\$2charSalt\$53charEncryptedpassword:16549:0:99999:7::  
katz3: \$2y\$2charSalt\$53charEncryptedpassword:16550:0:99999:7::
- ◆ **f1** Unique username (userid)  
**f2** Encrypted password (60 characters) `$2y$`-blowfish encryption algorithm used followed by `$2 Char Salt$` then `69 Char encryption` [note: different encryption algorithms have different lengths]  
**f3** Days since password was changed (0 = Jan 1, 1970)  
**f4** Days before password can change  
**f5** Days after which password must change (99999 = 273 yrs)  
**f6** Days before password will expire to start warning  
**f7** Days after password expires that account is disabled  
**f8** Days since account disabled (0 = Jan 1, 1970)
- ◆ Use to view shadow information `$ sudo chage -l <userid>`  
Reference: `man 5 shadow`



# /etc/group Format U2

- ◆ `$ echo " f1 f2 f3 f4"; grep 'katz' /etc/group`  
`f1 f2 f3 f4`  
`at:x:25:katz`  
`man:x:62:katz`
- ◆ `f1` Unique Group name  
`f2` placeholder for encrypted password (unused)  
`f3` Primary Group id (GID) Number (agrees with `f4` in `/etc/passwd`)  
`f4` userid memberships (comma separated)
- ◆ Reference: `man 5 group`



# /etc/default/useradd

- ◆ 

```
$ cat /etc/default/useradd
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1 #(don't disable when pw expires)
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
GROUPS=
UMASK=022
NO_GROUPS=true #(don't create group=userid)
```



# User Defaults

- ◆ member of **users** group GID=100
- ◆ home directory under [/home](#)
- ◆ Don't disable account when password expires
- ◆ starting shell is bash
- ◆ starter files
- ◆ mail spool creation
- ◆ permission mask owner: read, write, execute (access)  
group and others: read only
- ◆ Additional groups none
- ◆ Don't create group name = user name
- ◆ See [/etc/default/useradd](#) file to admire or change



# /etc/login.defs

- ◆ **/etc/login.defs - Configuration control definitions for the shadow package**
- ◆ **Sets up the appropriate environment for new users**
- ◆ **Selected Properties to that can be (re)set:**
  - MOTD\_FILE** filename for mesgs shown after login
  - ENV\_PATH** defines \$PATH=searchable list of dirs
  - PASS\_MAX\_DAYS, PASS\_MIN\_DAYS,**
  - PASS\_WARN\_AGE** default password expirations
  - UID\_MIN** first UID to use when creating new users
  - CREATE\_HOME** yes/no re home directory creation
  - USERGROUPS\_ENAB** yes = create private group  
no = all users belong to “users” groupname



# New User Exercise U2

1. Type **sudo cp -p /etc/login.defs /etc/login.defs-**
2. Type **sudo vim /etc/login.defs** to verify needed settings. Look for parameter **CREATE\_HOME** and insure it is **yes**; look for **USERGROUPS\_ENAB** insure it is **no**. Make the changes, type **:wq** Optional: **sudo vimdiffs /etc/login.defs\*** , (**:q!** twice)
3. Type **cd /etc/skel** to make that your current directory. Type **mkdir Pictures Documents** . Change contents of **.bashrc** to append the line: **export EDITOR=/usr/bin/vim** . To do this, type **sudo echo "export EDITOR=/usr/bin/vim" >> .bashrc**
4. Type **sudo /usr/sbin/useradd -m -d /home/sharon sharon ;**  
type **sudo passwd sharon** (specify the password: **12345678** )
5. Type **sudo passwd -n 30 -w 3 -x 90 sharon ;**  
type **sudo chage -l sharon** to view the results; type **su - sharon**



# New User Exercise 2 u1

6. Type **sudo groupadd devproject; groupadd accounting**
7. Use **usermod** to assign sharon user belong to devproject and to accounting via: **sudo usermod -A devproject, accounting sharon (or katz, if it exists) ;**
8. Type **id sharon;** to verify sharon has been correctly added to the group devproject and lists all the groups for that userid.



# User Creation Bash Script

```
#!/bin/bash
USAGE="Usage: ./adduser.bash # As root"
# Script to create (add) a new userid with password on SUSE Linux
#
if [ $(id -u) -eq 0 ]; then
    read -p "Enter username : " username
    read -s -p "Enter password : " password
    egrep "^$username" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "$username exists!"
        exit 1
    else
        pass=$(perl -e 'print crypt($ARGV[0], "password")' $password)
        useradd -m -d /home/$username -p $pass $username
        [ $? -eq 0 ] && echo "User has been added to system!" || echo "Failed\
to add a user!"
    fi
else
    echo "Only root may add a user to the system"
    exit 2
fi
```



# newusers Batch Command U2

- ◆ **newusers (8)** updates and creates sets of new users in a batch command
- ◆ Form: **\$ newusers [option(s)] [file]**
- ◆ file or STDIN is of the format:  
**newuserid:unencrypted pw:uid #:gid #:Comment:  
Home Dir:ShellToLaunch**
- ◆ Creates home directories but no files within (except .xauth).  
Populate these in a loop via:  
**newgroup=users  
for newuser in \$(cat newuserfile)  
do  
sudo cp -R /etc/skel ~\$newuser  
sudo chown -R \$newuser:\$newgroup ~\$newuser  
done**



# Removing Users

- ◆ userdel calls 3 localized scripts in **/usr/sbin**:  
**userdel.local**,  
**userdel-pre.local**, [remove cron and systemd user jobs]  
**userdel-post.local** [rebuild NIS database]
- ◆ Checklist:
  - Remove from any local user databases, phonelists
  - Remove from `/etc/aliases` or add forwarding address
  - Remove user's crontab file, pending at jobs, print jobs
  - Terminate user processes
  - Verify removal of user from `passwd`, `shadow`, `group` files
  - Remove (or relocate) user's home directory
  - Remove user's mail spool directory
  - Clean up entries on shared calendars, room reservation systems
  - Delete or transfer ownership of any mailinglists run by the user
  - Verify ex-UID files gone: **sudo find <filesystem> -xdev -nouser**



# password Strength

- ◆ **Password Strength** is a measure of how effective your password is at resisting guessing and brute-force attacks.
- ◆ It is a function of length, complexity and unpredictability.
- ◆ Some systems will impose a time-out for 3 failed password entry attempts.



# password Strength (2) u2

- ◆ **command-not-found** is in Repository: openSUSE-13.2-Oss, Version: 0.1.0-58.1.4 as a shell extension
- ◆ If possible, download **\$ cnf pwgen # or command-not-found pwgen**  
**\$ zypper install pwgen**  
to produce pseudo-random, pronounceable, passwords of varying number, length, character makeup
- ◆ Run **\$ pwgen -y -N 6 12 # use -s for greater randomness**
- ◆ Reference: <[https://en.wikipedia.org/wiki/Password\\_strength](https://en.wikipedia.org/wiki/Password_strength)>  
Also see <<https://xkcd.com/936/>>



# password Managers

- ◆ **Tokens** to protect credentials [**what you have**: smart card, USB stick; **what you know**: PIN or password; **What is part of you**: fingerprint, retina, face, hand ]
- ◆ **Stateless**: passwords generated on the fly
- ◆ **Characteristics**: Browser import, Competitor import, 2-factor authentication, Portable, Export data, Auto password capture, replay, fill in forms, pw Strength report, logins menu, Secure sharing
- ◆ **Defends against**: • phishing, • keyloggers • fake/real web site compare • limits to 3 consecutive failed password entries
- ◆ Doesn't protect stored password files; local much safer than browser based managers, autofill can be blocked by browsers.



# password Managers (2)

- ◆ Linux Password Managers with Browser Integration:
  - **Encryptr** (GPLv3) [Browser autotyping] cloud-based
  - **KeePass, KeePassX** (GPLv2) [Browser autotyping] local install
  - **Keeper** (Proprietary) local install+ cloud sync
  - **Password Safe** (Artistic License 2.0) [Browser autotyping] local install
- ◆ See:
  - <[https://en.wikipedia.org/wiki/Password\\_manager](https://en.wikipedia.org/wiki/Password_manager)>,
    - <[https://en.wikipedia.org/wiki/Comparison\\_of\\_password\\_managers](https://en.wikipedia.org/wiki/Comparison_of_password_managers)>,
      - <[https://en.wikipedia.org/wiki/List\\_of\\_password\\_managers](https://en.wikipedia.org/wiki/List_of_password_managers)>