Automatizar tareas de administración del sistema usando trabajos programados

Peso	4
Tópico Cubierto	107.2 Automatizar tareas de administración del sistema usando trabajos programados.
Descripción	Los candidatos deberán ser capaces de usar cron o anacron para ejecutar trabajos a intervalos regulares y usar at para ejecutar trabajos en una fecha específica.
Temas	Administrar trabajos de cron y at Configurar el acceso de usuario a servicios cron y at
Ejemplos	* /etc/cron.{d, daily, hourly, monthly, weekly} * /etc/at.deny * /etc/at.allow * /etc/crontab * /etc/cron.allow * /etc/cron.deny * /var/spool/cron * crontab * at * atq * atrm * anacron * /etc/anacrontab

Peso: Indica el valor de importancia que tiene este tópico en la certificación.

Tópico Cubierto: Indica, según el programa de certificación LPI, qué tópico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se deben tener en cuenta.

Introducción

En este tópico veremos temas relacionados con la automatización de tareas, tema fundamental para poder delegar tareas al sistema y controlar mejor su funcionamiento.

Utilizaremos y configuraremos servicios fundamentales como cron, at y anacron, viendo su funcionamiento y configuración en cada caso particular.

¿Qué son las tareas programadas?

En nuestro sistemas hay tareas que se realizan sin que nosotros tengamos intervención alguna. Esto es una manera de poder automatizar tareas y dejar de tener que estar presente para poder realizarla.

Para poder lograrlo necesitamos de un servicio llamado cron.

Utilizando Crontab

Para poder utilizar este tipo de tareas programadas primero debemos ver cómo es el archivo de configuración, para así programar nuestras tareas.

Las tareas las pueden programar los usuarios o también el sistema.

cat /etc/crontab

SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

For details see man 4 crontabs

Ejemplo de definición:

.---------------- minuto (0 - 59)

| .------------- hora (0 - 23)

| | .---------- día del mes (1 - 31)

| | | .--------- mes (1 - 12) o en inglés jan,feb,mar,apr ...

| | | | .------- día de la semana (0 - 6) (Domingo=0 or 7) OR sun,mon,tue,wed,thu,fri,sat

| | | | | .---- usuario a ejecutar la tarea (el usuario solo se configura en este archivo)

* * * * * usuario comando_a_ejecutar

SHELL	Indica que interprete deberá ejecutar los comandos
PATH	Estableceremos el camino de búsqueda de comandos que deberá seguir el sistema.
MAILTO	Se define el email al que llegarán las salidas de los comandos ejecutados.

Estas variables no son estrictamente necesarias.

Estas líneas muestra la distribución de la información en el archivo de configuración del crontab del sistema.

Comando crontab

Con el comando crontab podremos realizar ciertas tareas de administración, limitadas al usuario que las realice. A continuación, crearemos un crontab para un usuario normal del sistema.

Opciones
-l Lista las tareas programadas del usuario
-e Edita las tareas programadas
-r Borra las tareas programadas

Listar las tareas programadas

crontab -l
0 4 * * * /backup/backup.sh

En este ejemplo se ejecutará el script backup.sh todos los días a las 04:00
Ejecutar una tarea cada 5 minutos

El primer campo que corresponde a los minutos tiene / ,eso significa que lo hará cada 5 minutos (*/5)

*/5 * * * * /home/crond1/backup.sh

Nota: De la misma forma, usando */6 sería cada 6 minutos, */15 para 15 minutos, etc.

Ejecutar una tarea cada 5 horas

El segundo campo que corresponde a la hora tiene / ,eso significa que lo hará cada 5 horas (*/5)

0 */5 * * * /home/crond1/backup.sh

Nota: De la misma forma, usando */2 sería cada 2 horas, */3 para 3 horas, etc.

Ejecutar una tarea cada una hora de 9 a 18
Se ejecutará al minuto cero, de cada hora definida en el rango

0 9-18 * * * /home/crond1/backup.sh
Ejecutar una tarea el día 1 del mes

0 5 1 * * /home/crond1/backupfull.sh
Ejecutar todos los viernes a la medianoche

0 0 * * 5 /home/crond1/backup.sh

También se puede escribir el nombre del día en inglés

0 0 * * Fri /home/crond1/backup.sh

También podríamos utilizar algunos de los siguientes valores que corresponden a cada día.

0=Sun (También se puede utilizar el 7)

1=Mon

2=Tue

3=Wed

4=Thu

5=Fri

6=Sat

Ejecutar una tarea cada 5 meses

No hay una forma de decir exactamente cada 5 meses; para lograr eso debemos especificar en qué meses queremos correr la tarea donde, por ejemplo, el quinto mes sea mayo (May) y el décimo mes Octubre (Oct), separados por coma
0 0 * 5,10 * /home/crond1/backup.sh
También se puede escribiendo los nombres en inglés
0 0 * may,oct * /home/crond1/backup.sh

El siguiente ejemplo ejecutará la tarea dos veces al año, una el primero de Mayo a la medianoche, y otra, el primero de Octubre a la medianoche.

0 0 1 5,10 * /home/crond1/backup.sh

También se puede escribiendo los nombres en inglés

0 0 1 May,Oct * /home/crond1/backup.sh

Nota: No confundir 5,10 (Ejecutará en Mayo y Octubre) con 5-10 (Ejecutará desde Mayo hasta Octubre)

Ejecutar una misma tarea dos veces en el día

El siguiente script realizará un backup incremental dos veces al día, todos los días.

Este ejemplo ejecutará un shell script de backup incremental (incrementa-backup) a las 11 y a las 16, todos los días. La separación con la coma en el campo específico significa que necesita ser ejecutado en la hora correspondiente.

00 11,16 * * * /home/crond1/bin/incremental-backup

El siguiente ejemplo chequea el estado de la base de datos todos los días de semana (excluyendo los Sábados y Domingo) durante la franja horaria 9 a 18.

0 09-18 * * 1-5 /home/ramesh/bin/check-db-status

Atajos

En lugar de especificar los valores en los cinco campos, se puede especificar un “@” seguido de una palabra clave - tales como reboot, midnight, daily, monthly, yearly, hourly.

Clave	Equivalente
@yearly	0 0 1 1 *
@monthly	0 0 1 * *
@daily	0 0 * * *
@hourly	0 * * * *
@reboot	Arranca en el inicio.

Ejecutar el script mantenimiento_anual.sh de manera anual
@yearly /home/educacionit/bin/mantenimiento_anual.sh

Ejecutar comando borra_temporales.sh luego de reiniciar
@reboot borra_temporales.sh

Administración de tareas programadas

Los crontabs de los usuarios se encuentran en el directorio /var/spool/cron/ o /var/spool/cron/crontabs/ según la distribución.

Dentro de este directorio aparecerá un archivo con el nombre del usuario, dentro tendrá las tareas programadas

ls-l /var/spool/cron
-rw------- 1 crond1 root 523 dic 18 02:38 crond1

Editar las tareas programadas del usuario crond1
crontab -u crond1 -e

Listar las tareas programadas del usuario crond1
crontab -u crond1 -l

Borrar las tareas programadas del usuario crond1
crontab -u crond1 -r

Crontab del sistema

El archivo de cron de sistema es /etc/crontab. Si ponemos tareas en dicho archivo serán ejecutadas igualmente, aunque no es recomendable. Este crontab se deja para que lo maneje la distribución y sus programas. Es igual que el crontab de root, salvo que en este podemos especificar con qué usuario se ejecuta cada cosa, y cron hará una suplantación previa a la ejecución.

Archivos de Configuración

Los archivos de configuración del crontab se encuentran en /etc/cron.* También tenemos un /etc/crond.deny para denegar o un /etc/cron.allow para permitir el uso de cron.

ls -ld /etc/cron.*

drwxr-xr-x. 2 root root 4096 dic 18 04:24 /etc/cron.d

drwxr-xr-x. 2 root root 4096 feb 8 2011 /etc/cron.daily

-rw-r--r-- 1 root root   0 jun 29 09:55 /etc/cron.deny

drwxr-xr-x. 2 root root 4096 feb 8 2011 /etc/cron.hourly

drwxr-xr-x. 2 root root 4096 feb 8 2011 /etc/cron.monthly

drwxr-xr-x. 2 root root 4096 feb 8 2011 /etc/cron.weekly

Directorios predefinidos: hourly, daily, weekly y monthly

Los sistemas UNIX modernos vienen con directorios predefinidos para que cron lea y ejecute lo que hay dentro en los intervalos que su nombre indica:

/etc/cron.daily

/etc/cron.hourly

/etc/cron.weekly

/etc/cron.monthly

Dichos directorios se suelen utilizar para enlazar scripts que deben ser llamados en el intervalo correspondiente al directorio, sin argumentos. Por ejemplo, si creamos un script de bash y lo guardamos en /root/bin, le damos permisos de ejecución, y lo enlazamos en /etc/cron.hourly y cron lo ejecutará cada hora:

chmod +x /root/bin/miScript.sh

ln -s /root/bin/miScript.sh /etc/cron.hourly/

Algunos archivos importantes que deniegan el acceso a crontab son /etc/cron.deny, /etc/cron.allow; con estos dos archivos, dependiendo cuál utilicemos, le permitirán a los usuarios poder usar crontab, o, sino, especificar quiénes no van a poder utilizarlo. Es más útil definir quién lo puede utilizar, así acotamos el margen de error.

Servicio Crontab

Verificar el estado del servicio cron
service crond status

Redirecting to /bin/systemctl status crond.service

crond.service - Command Scheduler

Loaded: loaded (/lib/systemd/system/crond.service)

Active: active (running) since Sun, 18 Dec 2011 04:31:23 -0300; 5s ago

Main PID: 17885 (crond)

CGroup: name=systemd:/system/crond.service

└ 17885 /usr/sbin/crond -n

Reiniciar el servicio de cron (no es necesario reiniciar al agregar una tarea)
service crond restart
Restarting periodic command scheduler: cron [ok]
Starting periodic command scheduler: cron. [ok]

Tareas con AT

Con el comando at se pueden ejecutar trabajos por lotes, para ser ejecutados por única vez.

Se puede programar un trabajo de dos maneras diferentes:

- Programar el trabajo a ser ejecutado en un momento determinado. Por ejemplo, 03 de julio, 10am
- Programar el trabajo a ser ejecutado en el tiempo con respecto a partir de ahora. Por ejemplo, 5 horas a partir de ahora.

Tipo de Referencia	Sintaxis	Descripción
Fija	HH:MM	Específica la hora exacta y minutos cuando los comandos se debe ejecutar. El demonio at asume que la hora y el minuto especificado es hoy, a menos que el tiempo ha pasado realmente y asume que es mañana. También le podemos agregar am o pm para especificar mañana o tarde.
	Noon	Específica que el comando correrá a las 12:00 PM
	Midnight	Específica que el comando correrá a las 12:00 AM
	Teatime	Específica que el comando correrá a las 4:00 PM
	MMDDYY - MM/DD/YY - MM.DD.YY	Específica el mes , día y año exacto en que un comando se ejecutará
	HH:MM MMDDYY	Específica el mes , día, año y tiempo exacto en que un comando se ejecutará
Relativa	now	Específica que un comando debe ejecutarse inmediatamente
	now + valor	Específica que un comando debe ejecutarse en el futuro, en un tiempo estimado. Por ejemplo, now + 5 minutes, now + 2 hours, now + 3 days
	today	Específica que el comando debe correr hoy. También se lo puede usar con otros parámetros para determinar más opciones. Por ejemplo, 2 pm today
	tomorrow	Específica que el comando deberá correr mañana. También se lo puede usar con otros parámetros para determinar más opciones. Por ejemplo, 2 pm tomorrow

Ejemplos

1. Programar un trabajo at especificando fecha y hora

Syntaxis:

$ at tiempo fecha -f archivo_comandos

Programar un trabajo a las 11 am el 20 de mayo, hacemos lo siguiente siguiente:

$ at 11 am may 20 -f lista_de_comandos.txt

2. Programar un trabajo con at usando un tiempo relativo

Se puede programar un trabajo a ser ejecutado con el tiempo relativo a partir de ahora.

Syntaxis:

$ at now + Cantidad Unidad

Programar un trabajo para que sea ejecutado en un minuto a partir de ahora.

$ at now + 1 min -f lista_de_comandos.txt

Programar un trabajo para que sea ejecutado en un minuto a partir de ahora por STDIN

$ echo ”killall httpd” | at now + 1 min

Se puede programar un trabajo para que se ejecute en segundo plano a partir de una hora o un día, con el siguiente comando:

$ at now + 1 hour

$ at now + 1 day

Ingresar los comandos por medio del teclado, al terminar de tipear pulsar las teclas “ctrl d” y aparecerá <EOT> y la fecha en la que se ejecutará

at now + 2 min

at> touch /tmp/lala.txt

at> touch /tmp/lala2.txt

at> <EOT>

job 11 at Wed Dec 1 22:21:00 2014

Comando atq

El comando atq se utiliza para mostrar todos los trabajos en el orden que están programadas o en ejecución actualmente. Mostrará una lista de todos los trabajos pendientes. El primer número que se muestra es el número de trabajo, seguido de la hora en que el proceso se va a ejecutar, y el nombre del usuario. También se puede utilizar at -l

$ atq

4 2010-04-20 11:00 educacionit

Comando atrm

El comando atrm se utiliza para eliminar un trabajo en particular. Por ejemplo, para eliminar el trabajo número 4, utilizamos el siguiente comando atrm. También se puede utilizar atd -d 4

$ atrm 4

Comando batch

El comando batch ejecutará un trabajo sólo cuando el promedio de carga del sistema sea menor a 1,5

Al igual que el comando at se puede ejecutar batch ingresar los comandos y luego pulsar las teclas “ctrl d”

Actualizar la base de datos de locate cuando la carga sea menor a 1.5

$ batch

at> updatedb

at> <EOT>

job 11 at Wed Dec 1 22:25:00 2014

Archivos at.allow y at.deny

En primer lugar, el sistema controla el archivo at.allow. Si at.allow existe, sólo los nombres de usuario especificado en el archivo at.allow están permitidos para el uso de trabajos.

A continuación, (si at.allow no existe), se controla a at.deny. Si at.deny existe, a los nombres de usuarios especificados en el archivo at.deny no se les permite utilizar el comando at.

De manera predeterminada, la mayoría de los sistemas utiliza at.deny para poner fin al uso de trabajos a ciertos usuarios de sistema, como www-data, nobody, backup, etc.

Ejecutar un comando comando y luego salir de la shell

Podemos ejecutar un comando (o shell script) en el servidor remoto utilizando el comando at y salir de la shell.

$ at -f myjob now + 1 min

$ exit

Nota: myjob seguirá funcionando incluso después de salir fuera del servidor, de manera similar al comando nohup.

Otros tipos adicionales de formatos de hora del comando at

Puede utilizar cualquiera de los siguientes formatos en orden de fecha y hora:

$ at 10 am tomorrow

$ at 11:00 next month

$ at 22:00 today

$ at now + 1 week

$ at noon

Anacron

Anacron fue originalmente diseñado para ejecutar tareas en sistemas que no están encendidos todo el dia como por ejemplo PC de escritorio. Asegurando que la tarea programada, si es que no se ejecutó, se ejecute al iniciar el equipo.

Anacron corre por medio de cron vía el archivo 0anacron en /etc/cron.hourly. Por ende Anacron no corre como un demonio por su cuenta.

La configuración se encuentra en el archivo /etc/anacrontab y este ejecuta lo que se encuentra en los directorios /etc/cron.{daily,weekly,monthly}

Ejemplo:

SHELL=/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin

```
MAILTO=root

# the maximal random delay added to the base delay of the jobs

RANDOM_DELAY=45

# the jobs will be started during the following hours only

START_HOURS_RANGE=20-22

#period in days   delay in minutes   job-identifier   command

1         5         cron.daily              nice run-parts /etc/cron.daily

7         25        cron.weekly             nice run-parts /etc/cron.weekly

@monthly 45        cron.monthly            nice run-parts /etc/cron.monthly
```

El archivo /etc/anacrontab tiene los trabajos de anacron mencionados en el siguiente formato.

Período Retraso Identificador Comando (period delay job-identifier command)

Campo 1: es el período de recurrencia: Este es un valor numérico que especifica el número de días.

1 - diario

7 - semanal

30 - mensuales

N - Este puede ser cualquier valor numérico. N indica el número de días

Nota: También podemos utilizar @monthly para un trabajo que debe ser ejecutado cada mes.

Campo 2: Es el retraso en minutos, es decir, un número X de minutos que anacron debe esperar antes de ejecutarse luego que la máquina arrancó.

Campo 3: Es el nombre identificador del archivo de trabajo. Debe ser único para cada puesto de trabajo. Estará disponible como un archivo bajo el directorio /var/spool/anacron. Este archivo contendrá una sola línea que indica la última vez que este trabajo fue ejecutado.

```
# ls  /var/spool/anacron/

test.daily

cron.daily

cron.monthly

cron.weekly
```

```
# cat /var/spool/anacron/test.daily

20130507
```

Campo 4: Es el comando o shell script que debe ejecutarse.

Al igual que los scripts de shell, los comentarios dentro del archivo anacrontab empieza con #

START_HOURS_RANGE

¿Qué sucede cuándo la computadora portátil o de escritorio se apaga? ¿Cuándo se ejecuta el trabajo ? Esto se especifica mediante la variable de entorno START_HOURS_RANGE en el archivo /etc/anacrontab.

Por defecto se establece en 3.22 en el archivo. Esto indica que las tareas se ejecutarán entre la hora 3:00 y las 22:00.

grep START /etc/anacrontab

START_HOURS_RANGE = 3.22

RANDOM_DELAY

Anacron también añade un número x de minutos al azar. La x es definida por la variable RANDOM_DELAY en el archivo /etc/anacrontab.

Por defecto se establece 45 en el archivo. Esto significa que anacron sumará x minutos (elegidos al azar entre 0 y 45), y añadirá esto a la demora definida por el usuario.

Ejemplo

Ejecutar /home/educacionit/backup.sh una vez cada 7 días.

El día en que se supone que será ejecutado el trabajo backup.sh, si el sistema está apagado por algún motivo, anacron ejecutará el script backup.sh 15 minutos después de que el sistema vuelva a estar activo (sin tener que esperar por otros 7 días).

cat /etc/anacrontab

7 15 test.daily /bin/sh /home/educacionit/backup.sh

Cron Vs Anacron

Diferencias	* Anacron	Crond
	granularidad mínima es sólo de día	Granularidad mínima es de minutos (es decir, los trabajos se pueden programar para que se ejecuten cada minuto)
	Anacron puede ser utilizado sólo por super usuario (pero hay soluciones para que sea utilizable por el usuario normal)	Cron job pueden ser programadas por un usuario normal (si no se limita por super usuario)
	Anacron no espera que el sistema se ejecute 24 x 7. Si un trabajo está previsto, y el sistema esta apagado durante ese tiempo, el trabajo se ejecutara cuando el sistema vuelve a estar arriba.	Cron espera que el sistema se ejecute 24 x 7. Si un trabajo está previsto, y el sistema esta apagado durante ese tiempo, el trabajo no se ejecuta.
	Ideal para equipos de escritorio y portátiles	Ideal para servidores
	anacron se utiliza cuando un trabajo tiene que ser ejecutado independientemente de la hora y los minutos	Use cron cuando un trabajo tiene que ser ejecutado a una hora determinada y minutos

Integración

En la actualidad varias distribuciones, integran anacron y cron. Es decir vienen con el servicio anacron ya incorporado el cual llama de manera predeterminada a los scripts /etc/cron.{daily,weekly,monthly}. Esto sucede por ejemplo en CentOS 7.

Bibliografía

Editar la bibliografia esta mal apuntada

Libros:

LPI Linux Certification in a Nutshell, Third Edition, June 2010 —>Capitulo 7

LPIC-1: Linux Professional Institute Certification Study Guide: (Exams 101 and 102), 2nd Edition, February 2009 —>Capitulo 4

Páginas:

[1] Crontab

[2] Ejemplos Crontab

[3] Mas Ejemplos

[4] Ejemplos at

[5] Instalar crontab en Red Hat 6

[6] Anacron vs Cron

Localización e Internacionalización

Peso	3
Tópico Cubierto	107.3 Localización e Internacionalización
Descripción	Los alumnos deberían ser capaces de localizar un sistema en un idioma diferente al inglés. Además, un entendimiento de por qué LANG=C es útil cuando se escriben scripts.
Temas	• Configurar locale y variables de entorno • Configurar huso horario y variables de entorno
Ejemplos	• /etc/timezone • /etc/localtime • /usr/share/zoneinfo/ • LC_* • LC_ALL • LANG • TZ • /usr/bin/locale

	• tzselect • timedatectl • date • iconv • UTF-8 • ISO-8859 • ASCII • Unicode

Peso: Indica el valor de importancia que tiene este tópico en la certificacion.

Tópico Cubierto: Indica según el programa de certificacion LPI que topico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se tienen que tener en cuenta.

Introducción

Como muchos saben GNU/Linux es un sistema operativo internacional, con usuarios y colaboradores en diferentes lugares del planeta. Debido a eso, es muy importante el soporte a los distintos tipos de idiomas, caracteres, teclados, formato de fecha y tiempo , entre otras configuraciones regionales. Muchas de estas cuestiones se definen durante la instalación del sistema, pero nada impide realizarlo a posteriori, de esto último trata este tópico.

Configurando la zona horaria

Antes de empezar a hablar de cómo se configura tenemos que mencionar que nuestro sistema utiliza internamente UTC (Coordenada Universal de Tiempo), en donde refleja una hora estándar a nivel mundial en cualquier parte del mundo siempre es la misma hora, a lo que viene explicar cómo es esto. UTC viene por el uso de Greenwich (Royal Observatory, GMT) en donde todas las demás horas son calculadas desde ahí, existen lo que se llama husos horarios que determinan la hora correcta de cada país y teniendo el punto de referencia de Greenwich se dice que se esta a tantas horas de Greenwich. Para seguir explicando este tendríamos que volver a mencionar cómo funciona el reloj en nuestro sistema.

Relojes

Podríamos decir que antes de empezar a manejar la zona que nos corresponde podríamos mencionar como el sistema de tiempo funciona en nuestro sistema. Para eso tendríamos que mencionar que existen dos tipos de relojes en nuestro GNU/LINUX:

Hardware CLock: Este reloj funciona independientemente todo el tiempo, inclusive cuando la maquina está apagada. Se lo puede llamar de diferentes maneras, RTC (Real Time CLock) o BIOS/CMOS clock.

System Time: Este reloj es el que corre internamente dentro del Kernel de Linux, es manejado por una interrupción de tiempo ISA. La hora del sistema se mide como el número de segundos desde la 00:00:00 del 1/01/1970 UTC.

Algo importante a tener en cuenta es recordar que el hardware clock y el System Time no son el mismo. Cuando estemos administrando un sistema GNU/LINUX estaremos acostumbrados a manejar el System Time que el reloj hardware clock, el rol de este último es muy básico dado que solo es mantener el sistema de

tiempo cuando la maquina está apagada. El System Time es sincronizado con el hardware clock una vez que nuestro sistema arranca, luego del arranque se usará el System Time y el otro es ignorado.

Configurar el reloj del sistema a UTC es la opción más elegida cuando trabajamos con Linux dado que usando UTC podemos calcular más fácil la hora de nuestra zona. Por ejemplo si utiliza algún programa o desarrollo en donde va a ser accedido por diferentes puntos del mundo va a resultar mucho más fácil hacer la conversión de hora dependiendo de la zona que corresponda.

Reloj del sistema con zona correspondiente:

date

Wed Dec 10 23:08:31 ART 2014

Reloj del sistema con UTC

date --utc

Thu Dec 11 02:09:03 UTC 2014

Administrando Time Zone

Como mencionamos anteriormente cuando instalamos nuestro sistema tenemos que indicarle en qué zona estamos , si quisiéramos ver cómo configurarla de vuelta tendríamos antes que entender donde se encuentran algunos archivos.

Para eso tenemos varios archivos :

Archivos:

/etc/localtime

/etc/sysconfig/clock (si usa reloj sistema local o de hardware)

/etc/timezone (debian)

/usr/share/zoneinfo

Comandos:

tzselect

system-config-date

El archivo /etc/localtime por lo general es un link simbólico a la zona que corresponde que esta en /usr/share/zoneinfo

ls -l /usr/share/zoneinfo/America/Argentina/

lrwxrwxrwx 1 root root 32 Jun 2 2014 Buenos_Aires -> ../../posix/America/Buenos_Aires

lrwxrwxrwx 1 root root 29 Jun 2 2014 Catamarca -> ../../posix/America/Catamarca

lrwxrwxrwx 1 root root 29 Jun 2 2014 ComodRivadavia -> ../../posix/America/Catamarca

lrwxrwxrwx 1 root root 27 Jun 2 2014 Cordoba -> ../../posix/America/Rosario

(...salida cortada...)

Como ven se va dividiendo en cada directorio y al final esta el archivo que corresponde a la zona adecuada.

También podríamos ver este otro archivo en distros basadas en Red Hat.

cat /etc/sysconfig/clock

ZONE=”America/Argentina/Buenos_Aires”

Comando tzselect

El comando tzselect va a ayudar a seleccionar la zona horaria correcta

$ tzselect

Please identify a location so that time zone rules can be set correctly.

Please select a continent or ocean.

1) Africa

2) Americas

3) Antarctica

4) Arctic Ocean

5) Asia

(...salida cortada...)

#? 2

Please select a country.

1) Anguilla 28) Haiti

2) Antigua & Barbuda 29) Honduras

3) Argentina 30) Jamaica

4) Aruba 31) Martinique

5) Bahamas 32) Mexico

(...salida cortada...)

#? 4

The following information has been given:

Aruba

Therefore TZ='America/Aruba' will be used.

Local time is now: Wed Dec 10 22:22:59 AST 2014.

Universal Time is now: Thu Dec 11 02:22:59 UTC 2014.

Is the above information OK?

1) Yes

2) No

#? 1

You can make this change permanent for yourself by appending the line

TZ='America/Aruba'; export TZ

to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you

can use the /usr/bin/tzselect command in shell scripts:

America/Aruba

El comando nos muestra la variable a setear para la zona horaria seleccionada.

Setear la zona horaria en la terminal actual. Esto no afectará a las demás terminales.
$ TZ='America/Aruba'; export TZ

De esta manera
$ date
Wed Dec 10 22:24:49 AST 2014

La variable puede definirse en el archivo .bashrc dentro del home del usuario, para que en el próximo acceso ya inicie con la zona horaria definida.

echo “TZ='America/Aruba'; export TZ” >> ~/.bashrc

Crear un archivo, y cada usuario lo va a ver con la hora calculada según la zona horaria definida

$ ls -l pruebahora.txt

-rw-r--r-- 1 usuario1 usuario1 0 Dec 10 22:29 pruebahora.txt

$ ls -l pruebahora.txt

-rw-r--r-- 1 usuario1 usuario1 0 Dec 10 23:29 pruebahora.txt
Como ven ambos tienen horas distintas, pero como mencionamos anteriormente el sistema maneja la hora en UTC por lo cual lo único que hace es transformar la hora según la zona que tengamos configurada.

Cambiar la hora del sistema

Para poder cambiar la hora del sistema de forma global únicamente para el sistema tenemos que editar el archivo /etc/localtime para que apunte a otra zona.

Ver la hora actual:
date

Wed Dec 10 23:31:28 ART 2014

Borrar el archivo de la hora local

rm /etc/localtime

Establecer la nueva zona horaria

ln -s /usr/share/zoneinfo/Pacific/Rarotonga /etc/localtime

Verificar la hora nuevamente

date

Wed Dec 10 16:34:31 CKT 2014

Comando timedatectl

El comando timedatectl es una herramienta de systemd, sirve tanto para mostrar información como para configurar (tiempo local, tiempo universal, reloj de tiempo real, zona horaria, configuración y sincronización por NTP)

Para mostrar la información, simplemente se puede ejecutar el comando:

timedatectl

Local time: mar 2017-04-11 09:34:07 -03

Universal time: mar 2017-04-11 12:34:07 UTC

RTC time: mar 2017-04-11 12:34:07

Time zone: America/Argentina/Buenos_Aires (-03, -0300)

Network time on: yes

NTP synchronized: yes

RTC in local TZ: no

Podemos hacer que muestre un listado de las zonas horarias y encontrar la que necesitamos, por ejemplo:

timedatectl list-timezones | grep Buenos

America/Argentina/Buenos_Aires

Esta herramienta puede además, permite realizar operaciones remotas, como vemos a continuación:

timedatectl -H remote.host.example status

root@remote.host.example's password:

Local time: mar 2017-04-11 09:54:48 -03

Universal time: mar 2017-04-11 12:54:48 UTC

RTC time: n/a

Time zone: America/Argentina/Buenos_Aires (-03, -0300)

Network time on: yes

NTP synchronized: yes

RTC in local TZ: no

Hay que tener en cuenta que timedatectl no puede modificar la fecha y/u hora si se usa un servidor NTP:

timedatectl set-time '2017-04-11 09:59:50'

Failed to set time: NTP unit is active

Configuración regional

Cuando estamos en la instalación de nuestro sistema este es uno de los primeros puntos que realizamos porque configurar el idioma es lo primero que nos pregunta a la hora de la instalación, luego esta decisión acarrea otras configuraciones por defecto debido al idioma que elegimos, Podríamos enunciar algunos parámetros que configuramos como conjunto de caracteres, el formato de la hora, el formato del dia, el formato de moneda por defecto entre otras más.

Podríamos decir que la forma que va a tomar nuestra configuración sería la siguiente –> [language[_territory][.codeset][@modifier]]

Cada parte del parámetro mencionado arriba tiene sus valores a tomar. Por ejemplo para language vendría a tomar el valor del idioma en (English), fr(French), es(Spanish), etc. Generalmente son dos o tres letras para identificar el idioma.

Para territory podria ser US(United States, FR (France), JP (Japan),AR (Argentina) etc. Son códigos específicos para cada región.

Para codeset podría ser ASCII, UTF-8 u otras codificaciones. El método de codificación ASCII (American Standard Code for Information Interchange) es la mas vieja y primitiva de todas, soporta codificación de 7-bit (generalmente almacenada en 8-bytes) en donde puede manejar la codificación en inglés incluyendo las puntuaciones y símbolos más comunes.

ASCII no puede manejar caracteres usados en varios idiomas que no sean el inglés, es por eso que no es usado internacionalmente. ISO-8859 fue creada para extender ASCII dado que usa un octavo bit para extender ASCII a 128 caracteres pudiendo así tener un soporte limitado para alfabetos no Romano. Posteriormente salerion ISO-8859-1 y ISO-8899-5 en donde se dió soporte a europa occidental y a Cirílico.

La codificación de lenguaje más usada es el Unicode Transformation Format de 8-bit (UTF-8). Como su predecesor ISO-8859 este tambien arranco con ASCII pero se extiende dando soporte por medio de un bit variable en donde un único caracter puede tomar desde uno a cuatro bytes para ser codificado y esto le provee la habilidad de codificar texto en cualquier idioma soportado por Unicode, por lo que puede soportar tantos lenguajes como sea posible. Otra ventaja importante sobre ISO-8859 es que no se necesito documentar otro subestándar (como ISO-8859-1 y 5). UTF-8 maneja todos sus sistemas de escritura de forma automática.

El modifier es un código específico de la localidad que se modifica la forma en que funciona. Por ejemplo, puede afectar el orden de clasificación de una manera específica del idioma.

Configuración

Ahora para poder configurar y ver cómo se encuentra nuestra codificación podríamos utilizar el siguiente comando:

locale

LANG=en_US.UTF-8

LANGUAGE=en_US:en

LC_CTYPE="en_US.UTF-8"

LC_NUMERIC="en_US.UTF-8"

LC_TIME="en_US.UTF-8"

LC_COLLATE="en_US.UTF-8"

LC_MONETARY="en_US.UTF-8"

LC_MESSAGES="en_US.UTF-8"

LC_PAPER="en_US.UTF-8"

LC_NAME="en_US.UTF-8"

LC_ADDRESS="en_US.UTF-8"

LC_TELEPHONE="en_US.UTF-8"

LC_MEASUREMENT="en_US.UTF-8"

LC_IDENTIFICATION="en_US.UTF-8"

LC_ALL=

Teniendo en cuenta lo visto anteriormente podemos entender un poco mejor acerca de los valores que tienen asignadas cada variable.

Variable	Descripción
LANG	Especifica el valor de idioma predeterminado para todas las LC_variables
LC_CTYPE	Configura el tipo de caracteres y codificación
LC_NUMERIC	Configura el formato de número
LC_TIME	Configura como se visualizara la fecha y la hora
LC_COLLATE	Configura las reglas de clasificación
LC_MONETARY	Configura el formato de moneda
LC_MESSAGES	Configura los mensajes del lenguaje natural
LC_PAPER	Configura el tamaño de papel predeterminado
LC_NAME	Configura el formato por defecto del nombre personal
LC_ADDRESS	Configura el formato de la dirección por defecto
LC_TELEPHONE	Configura el format del numero de teléfono por defecto
LC_MEASUREMENT	Configura la unidad de medida predeterminada
LC_ALL	Anula todas las otras variables de entorno LC
LANGUAGE	Anula a LC_MESSAGES

Si por alguna razón se necesita configurar la localización de nuestro sistema con un valor diferente al dado en la instalación tendríamos que cambiar simplemente las variables que correspondan al cambio que quisiéramos hacer.

Pueden tomar valores independientes si lo necesitamos pero hay que tener en cuenta que algunas modifican a las demás variables anulando sus valores.

1. Si la variable LC_ALL está definida anula todos las demás variables.

2. Si LC_ALL no está definida tendríamos que definir cada valor para las variables LC.

3. Si la variable LC es nula entonces el valor que se va a usar es LANG.

Con el comando locale también podemos ver listar los tipos de localización disponible:

```
# locale -a |grep AR
es_AR
es_AR.iso88591
es_AR.utf8
```

También se pueden ver los tipos de codificación disponibles:

```
# locale -m
ISIRI-3342

ISO-8859-1

ISO-8859-10
(...salida cortada...)
```

La variable LANG=C se utiliza en algunos scripts y evita problemas con la codificación, ya que utiliza ASCII puro.

```
# LANG=C; export LANG
# locale
LANG=C
LC_CTYPE=”C”
LC_NUMERIC=”C”
LC_TIME=”C”
(...salida cortada…)
```

Ejemplo:

Muestra el manual del man en español
```
# LANG=es_AR.utf8 man man
```

Muestra el manual del man en inglés
```
# LANG=en_US man man
```

Comando iconv

El comando iconv se utiliza para convertir un texto que tenga una codificación en otra.
Sintaxis

```
iconv [opciones] archivo
```

Opciones:
-f Define la codificación de origen
-t Define la codificación de destino
-o Archivo de salida

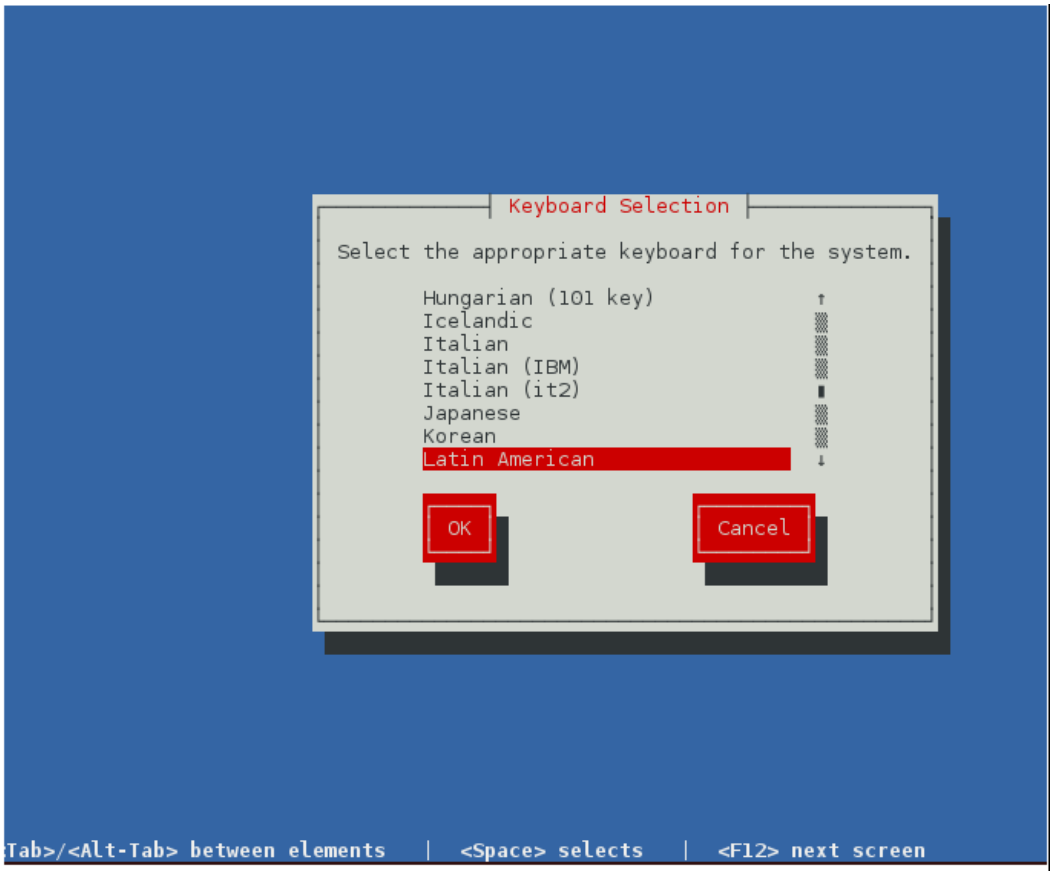
Cambiar de iso-8859-1 a UTF-8
```
# iconv -f iso-8859-1 -t utf-8 -o modificado.txt entrada.txt
```

Configuración de Teclado e Idioma

Basadas Red Hat

Cambiar Teclado

Comando system-config-keyboard



Con loadkeys

Para cuando estamos en una consola/terminal sin gráfica:

loadkeys es
Loading /lib/kdb/keymaps/xbd/es.map.gz

Tenemos que tener instalado el paquete kbd-misc para tener los mapas correspondientes.

Con setxkbmap

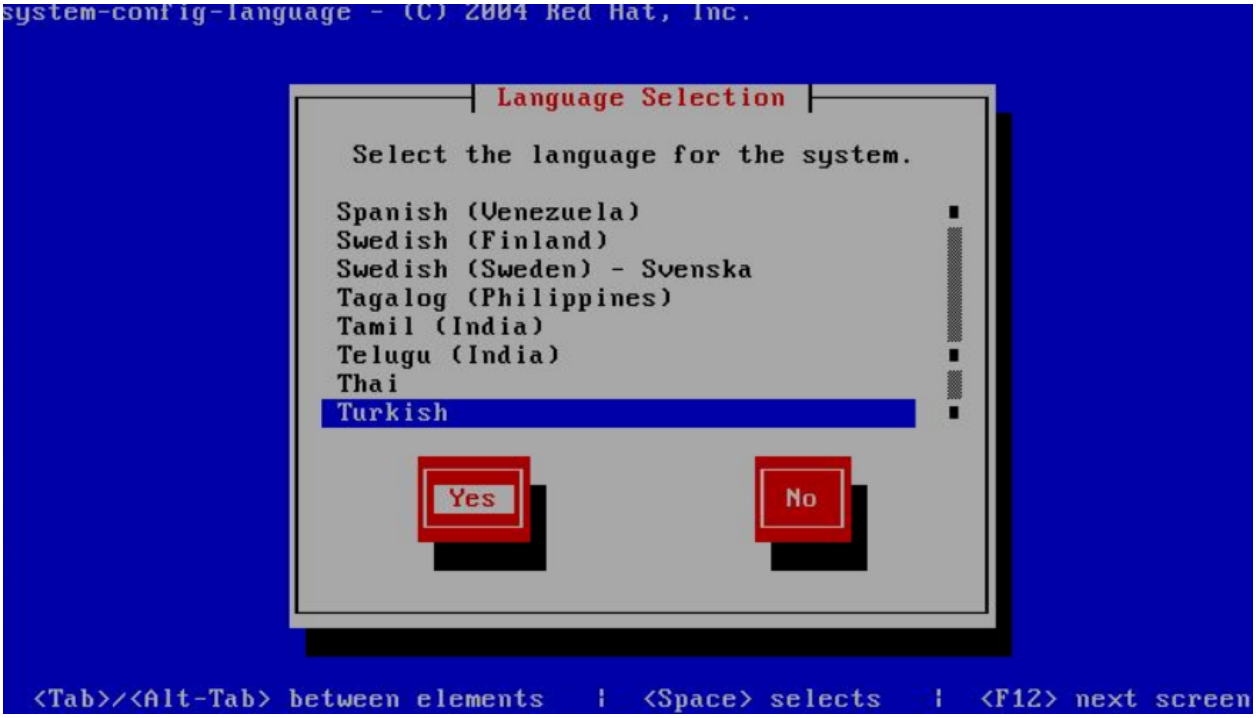
Con este comando se cambia el teclado desde una interfaz gráfica
setxkbmap latam

Cambiar Idioma

En las distribuciones basadas en Red Hat el archivo general se encuentra en:
# cat /etc/sysconfig/i18n
LANG=”es_ES.UTF-8”
SYSFONT=”latarcyrheb-sun16”

Con system-config-language

Para elegir el idioma del sistema



Basadas Debian

Cambiar Teclado

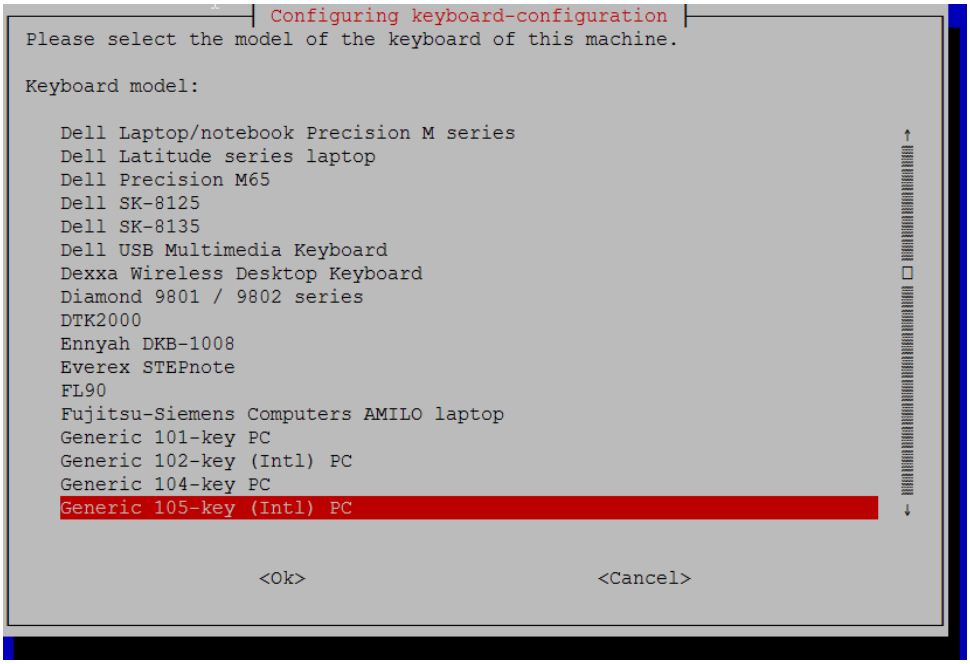
Veamos diferentes formas:

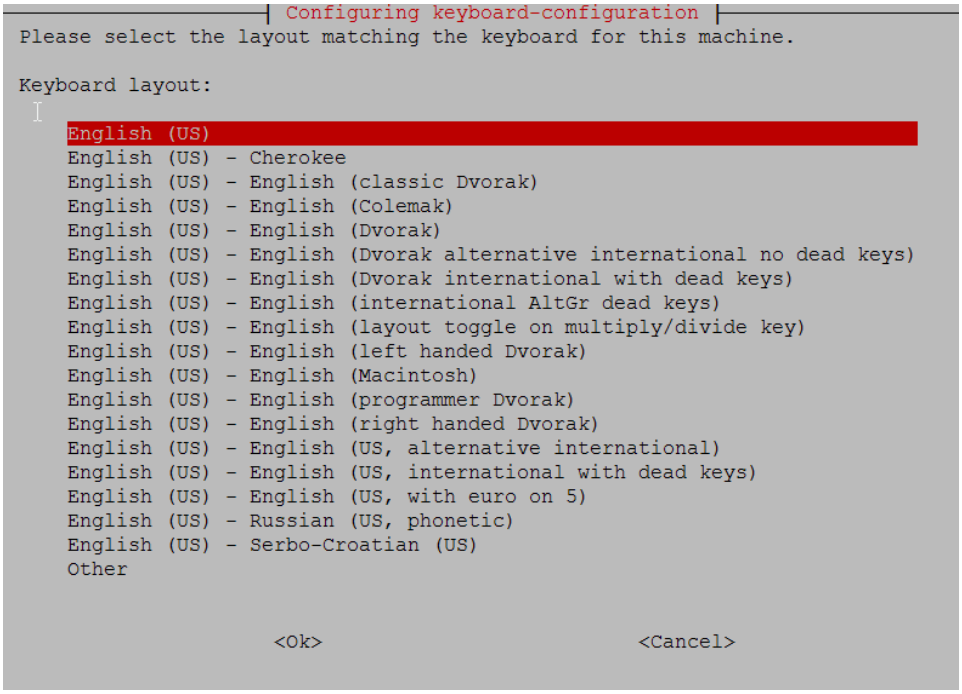
Con dpkg-reconfigure

Se utiliza el siguiente comando:

dpkg-reconfigure keyboard-configuration

En algunas versiones más viejas de debian:
dpkg-reconfigure console-data





Tendremos que seleccionar las opciones que nos correspondan.

Con loadkeys

loadkeys es
Loading /lib/kbd/keymaps/xkb/es.map.gz

Cambiar Idioma

Archivo de configuración general de idioma

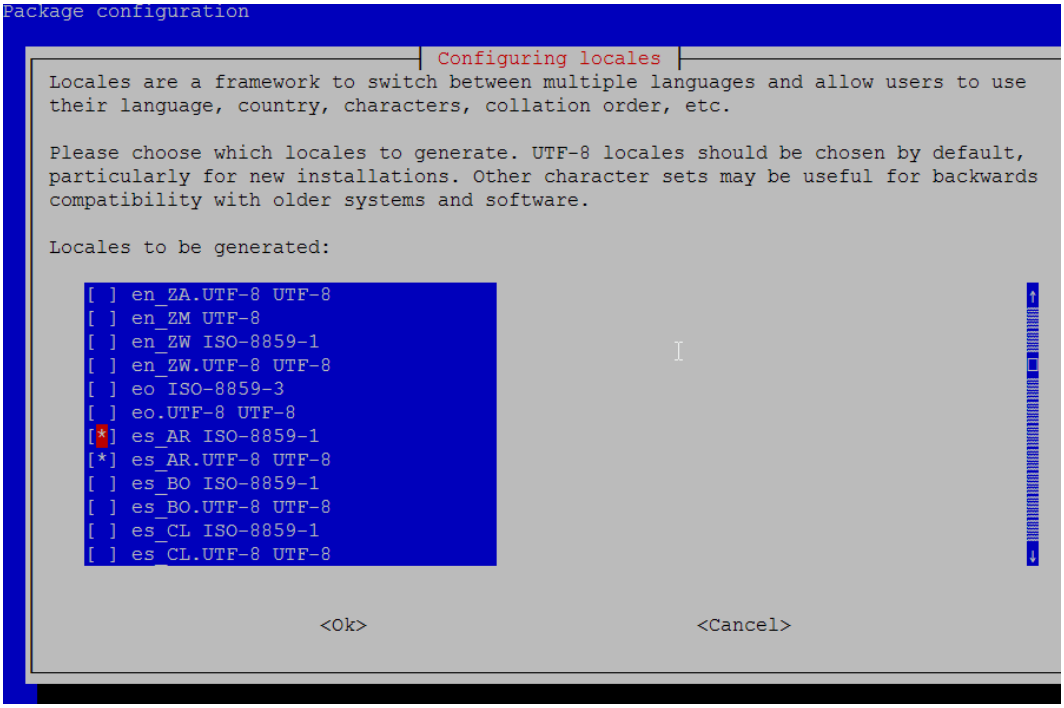
cat /etc/default/locale

LANG=en_US.UTF-8

LANGUAGE="en_US:en"

Con dpkg-reconfigure locales

Veamos su salida:



Con locale-gen

En el paso previo se configuró nuestro perfil, para hacerlo global corremos el script.

```
# locale-gen
Generating locales (this might take a while)...
 es_AR.UTF-8… done
Generation complete.
```

Bibliografía

Libros:

LPI Linux Certification in a Nutshell, Third Edition, June 2010

LPIC-1: Linux Professional Institute Certification Study Guide: (Exams 101 and 102), 2nd Edition, February 2009