Компютърът под контрол

4.1 Синхронизация на софтуера - версии, зависимости, библиотеки

Всяка Linux система можем да разглеждаме като множество от инсталирани пакети. Особено място заемат библиотеките - това са пакети които са предназначени за ползване от други пакети, зареждат се динамично и обикновено изпълняват услуги, изисквани от някакъв конкретен стандарт. Всеки нормален пакет при работата си използва една или няколко библиотеки, а често и услуги от други нормални пакети. Всяка дистрибуция на Linux се стреми да поддържа коректни зависимости между пакетите в следните направления:

- пълнота - всеки пакет се инсталира само ако библиотеките и останалите пакети от които зависи, са вече достъпни (инсталирани).
- непротиворечивост - ако два или повече пакета предизвикват конфликт при съвместна работа, дистрибуцията не допуска едновременната им инсталация.
- минималност - ако даден пакет не е необходим за директно ползване от собственика на компютъра и от него не зависят други пакети, той не се инсталира.

Обикновено конкретната дистрибуция поддържа специален файлов формат на пакетите си (*.rpm, *.deb), съдържащ цялата информация необходима за автоматизирано инсталиране/деинсталиране на пакета, така че да не се наруши синхронизацията на цялата система.

Случва се обаче дистрибуцията или пакета да има грешка, или пък да искате да инсталирате пакет, който все още не е адаптиран към дистрибуцията. В този случай се налага да инсталирате ръчно пакета. Следвайте такава последователност:

- Снабдете се със сорсовете (изходните текстове на пакета) и документация за него !
- Разберете от документацията от кои други пакети зависи !
- Проверете кои от тях са инсталирани ! Инсталирайте останалите !
- Чак сега компилирайте и инсталирайте мечтания пакет !

При проследяване на зависимостите често номера на версията има голямо значение. За всеки Linux пакет това е символен низ от вида xx.yy.zz, където:

- xx е основният номер на версията. 0 означава, че това е сравнително нов пакет, който все още не се ползва за отговорни задачи. 1 имат пакетите, достигнали стабилно състояние, а всеки следващ номер се дава при сериозна промяна във функционалността и стандартите.
- yy е номер на поредната реализация. За много пакети (традиция, идваща от ядрото) четна реализация означава стабилна версия, предназначена за ползване при сериозна работа, а нечетната е за разработване - за изпробване на нови услуги, за откриване и отстраняване на грешки и пр.
- zz е номер на кръпка (patch). Всеки пакет се изгражда чрез последователно извършване на сравнително малки промени. Тези промени се наричат кръпки. Ако разполагате с изходните текстове, можете да генерирате следващи версии само като изтеглите от Мрежата кръпките (обикновено те са на порядъци по-малки по обем) и ги залепите към старата си версия чрез командата patch.

4.2 Следене на състоянието

Колкото и автоматизиран да Ви изглежда вашият Linux, не е зле да го наглеждате от време на време и да сте наясно с методите за следене на текущото състояние. Те са толкова много и разнообразни, че за сравнително пълно изложение ще е необходим отделен учебник. Тук излагаме само най-често използваните и сравнително стари техники за контрол над компютъра, които могат да се използват от обикновена текстова конзола:

dmesg изписва съобщенията на ядрото при началното зареждане, както и някои последващи съобщения. Тази команда ще ви даде информация за услугите, които ядрото Ви поддържа, дали са открити вашите хардуерни устройства и дали при стартирането на ядрото всичко е наред.

syslogd е демон, който записва бележки, които различни програми му изпращат за да е достъпна историята на работата им. Файлът /etc/syslog.conf описва кои типове услуги къде ще насочват съобщенията си. Обикновено резултатите от работата на syslogd се записват в различни файлове от директорията /var/log и нейните поддиректории.

who дава информация за включените потребители.

ps изписва работещите в момента процеси.

netstat дава различни справки за състоянието на мрежовите услуги - TCP връзки, таблици за рутиране, състояние на мрейовите интерфейси и пр.

last дава списък на последните свързали се към системата потребители.

df дава статистика за използваната и свободна дискова памет по всички монтирани дискови дялове.

/proc е специална файлова система, отразяваща състоянието на ядрото. В различните поддиректории ще намерите информация за компонентите на вашата система, както и за състоянието на различните процеси. Няколко команди ползват тази информация и дават по-нагледна представа за компонентите:

- free за състоянието на RAM и swap паметта.
- top следи състоянието на процесите и системата изобщо.

ipchains е програма за управление и следене на мрежовите връзки. Взаимодействайки със специална част от ядрото, тя задава правила за работа с мрежовите пакети. Правилата позволяват спиране, пренасочване, преброяване, маскиране на пакетите. Най-често ipchains се ползва за следене и отчитане на трафика в мрежата, за забрана на нежелани мрежови услуги и за маскиране на intranet мрежа.

4.3 Добавяне и изключване на услуги

Linux е система, която работи толкова стабилно, че не се налага да бъде рестартирана

при добро поддържане. Това, както и разпределеното и малко координирано създаване на програми изисква сравнително детайлно познаване на механизмите за настройка на услугите, които ползвате на конкретния компютър. Допълнителна трудност е липсата на стандарт за стартиране на системата - всяка основна дистрибуция поддържа собствен метод за стартиране.

Временното (само за конкретната работна сесия) стартиране и спиране на програми е лесно - пускането става като напишем името на програмата, последвано от евентуални параметри. Можем накрая да сложим и символа '&', за да пуснем програмата във фонов режим. Спирането става с команда kill.

Значително по-труден за управление е процеса на стартиране на програми по време на началното зареждане на системата.

Общото между различните дистрибуции е началото на стартирането на Linux. То протича така:

- след зареждането и стартирането си ядрото стартира процеса init - единственият процес без родител.
- init прочита конфигурационния си файл /etc/inittab и започва да стартира системата съобразно информацията в този файл. Редовете в /etc/inittab имат вида 'етикет:ниво:действие:процес'.
- Първи се пуска процеса, указан в реда с действие 'sysinit'. Ето този ред за основните дистрибуции:
- SlackWare: si:S:sysinit:/etc/rc.d/rc.S
- RedHat:
- Debian: si::sysinit:/etc/init.d/rcS
- Caldera:

По нататък съответните скриптове съвместно с init започват да извършват дейностите по зареждане на системата, смяна на нивото и стартирането на услуги. По принцип за всяка дистрибуция скриптовете за стартиране и спиране на услуги са разположени на едно място, но в по-автоматизираните (RedHat, Debian) пускането на тези скриптове се извършва от символни линкове към тях, разположени в директории, съответстващи на нивото.

Тъй като материята е сложна, ще дам само един пример от дистрибуцията Debian. При тази дистрибуция скриптовете се намират в /etc/init.d и нека приемем, че скрипта за стартиране на SQL сървера ни е /etc/init.d/postgres. Тъй като този скрипт се пуска автоматично, той трябва да реагира на стандартен параметър, приемащ стойности:

- start - за стартиране на услугата
- stop - за спиране
- restart, install, uninstall и пр. - за други работи

Същинското пускане на скрипта обаче ще се извърши чрез символни връзки. В нашия конкретен случай искаме сървера да стартира на ниво 2, към края на стартирането на системата. За тази цел създаваме символна връзка (линк) в директория /etc/rc2.d така:

S95postgres -> ../init.d/postgres

Директорията /etc/rc2.d съдържа линкове към всички услуги, които ще бъдат пуснати или спрени при включване на ниво 2.

Буквата S в началото на името на връзката означава 'стартирай' (буква K означава 'спри').

Числото 95 в името на връзката влияе на реда на стартиране на услугите (00 се пускат първи, 99 - последни).

Тази наглед сложна схема за стартиране/спиране на услуги позволява унифициране на процеса на пускане/спиране на системата, както и методите за инсталация на пакети. Например ако искаме SQL сървера да не се стартира при рестарт на компютъра, просто изтриваме или преименуваме съответната символна връзка.

Единствено в дистрибуцията SlackWare процеса на стартиране се описва директно. Всички скриптове се намират в /etc/rc.d, а стартирането на многопотребителското ниво

се извършва от скрипта /etc/rc.d/rc.M, където пряко се викат всички услуги.

4.4 Индекс на важни системни команди

kill [-сигнал] номер_на_процес

Изпраща сигнал към указания процес, обикновено с цел да го спре. При липса на 'сигнал' се изпраща TERM. По твърдата форма е 'kill -9 pid'.

nice [-n приоритет] [команда [аргумент...]]

Стартира 'команда' с нов приоритет. Добавя 'приоритет' към приоритета на родителския процес. Стандартно процесите са с приоритет 0, -20 е най-висок, 19 е най-нисък, а nice добавя 10 при липса на аргумент 'приоритет'. Само 'nice' връща приоритета на текущия процес.

pidof процес

Връща номера на указания процес. Можете да спрете процеса my_proc като напишете "kill -9 `pidof my_proc`".

fuser файл или мрежова връзка

Връща номера на процес, който използва указания файл или мрежова връзка (socket).

ps параметри

Връща списък на активните процеси.