13. Потребители и групи  
• Файлове свързани с потребители и групи (passwd, group, shadow, gshadow);  
• Администриране на потребители и групи (добавяне, модифициране и изтриване);  
• Политики за пароли (chage) ;  
• Потребителски файлове по подразбиране (/etc/skel, /etc/profile) ;  
• Изпълняване на команди чрез sudo ;  
• Поставяне на ресурсни ограничения на потребителите (ulimit).

Достъпът до всяка Линукс система става чрез потребителски акаунт, установен от системния администратор. Това означава, че на всеки потребител се задава потребителско име за идентификация, парола, собствена директория и му се задават права за достъп до системата. Всичко това се прави чрез администраторския акаунт, наречен root.

Потребителя root се създава автоматично при инсталацията на дистрибуцията. Този потребител се нарича още свръхпотребител (суперпотребител), защото няма нещо, което този потребител да не може да направи. Пред този потребител няма никакви забрани или ограничения и затова работата с него е много удобна, но също толкова опасна. Поради значимостта на този потребител, командния интерпретатор променя обичайния prompt на командния ред.

Важно е да се отбележи, че името на този потребител не е задължително да бъде root, но това е името по подразбиране.

Освен суперпотребителя всяка Линукс система може да има и потребителски акаунти. Всеки акаунт има собствена директория в дървото на операционната система (изключение правят някои системни акаунти). Тя се нарича home (домашна) директория и потребителя попада винаги в нея след своя вход. Директорията се задава при създаването на потребителя и по подразбиране е : */home/ime\_na\_potrebitelia* .

Обикновено потребителя има пълни права в тази директория, но това не е задължително. Например може да се ограничат правата на потребителя върху определени файлове в тази директория. Класически пример за това е ограничаването на правата върху файла profile, който се изпълнява след включване на потребителя в системата.

Цялата информация за потребителите се съхранява във файла*/etc/passwd*. Това е най-обикновен текстов файл, чийто собственик е суперпотребителя. Само той може да редактира съдържанието му, а останалите потребители имат права само за четене. Редовете в този файл имат строго определен формат:

*username:password:userID:groupID:comment:home\_directory:login\_comm*

Нека разгледаме един примерен файл:

*root:x:0:0::/root:/bin/bash*

*bin:x:1:1:bin:/bin:*

*daemon:x:2:2:daemon:/sbin:*

*adm:x:3:4:adm:/var/log:*

*lp:x:4:7:lp:/var/spool/lpd:*

*sync:x:5:0:sync:/sbin:/bin/sync*

*shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown*

*halt:x:7:0:halt:/sbin:/sbin/halt*

*mail:x:8:12:mail:/:*

*news:x:9:13:news:/usr/lib/news:*

*uucp:x:10:14:uucp:/var/spool/uucppublic:*

*operator:x:11:0:operator:/root:/bin/bash*

*games:x:12:100:games:/usr/games:*

*ftp:x:14:50::/home/ftp:*

*smmsp:x:25:25:smmsp:/var/spool/clientmqueue:*

*mysql:x:27:27:MySQL:/var/lib/mysql:/bin/bash*

*rpc:x:32:32:RPC portmap user:/:/bin/false*

*gdm:x:42:42:GDM:/var/state/gdm:/bin/bash*

*pop:x:90:90:POP:/:*

*nobody:x:99:99:nobody:/:*

*sshd:x:33:33:sshd:/:*

*dino:x:1000:100:Dinko,,,:/home/dino:/bin/bash*

*www:x:1001:102:,,,:/home/www:/bin/bash*

*ffsearch:x:1002:103::/home/ffsearch:*

*stinger:x:1003:100:,,,:/home/stinger:/bin/rzsh*

*postfix:x:1004:100::/home/postfix:*

  Всеки ред от този файл е съставен от седем полета, разделени с двоеточия. Ако нищо не се въведе между две двоеточия полето остава празно, но двоеточията задължително трябва да съществуват. Ето и обяснение на седемте полета:

* username – уникалното потребителско име

* password – паролата на потребителя в кодиран вариант

* userID(UID) – уникалното число, което идентифицира потребителя пред системата

* groupID(GID) – уникално число, което определя групата на потребителя

* comment – коментар (може да истинското име на потребителя и т.н.)

* home\_directory – това е директорията в която попада потребителя след своето включване

* login\_comm – това е команда, която се изпълнява след вход на потребителя. Най-често това е някаква шел команда.

  Трябва да се отбележи, че тази структура на файла е еднаква не само за Линукс, но и UNIX базираните операционни системи.

Както бе казано по-горе, операционната система записва паролата във второто поле. Повечето съвременни версии, не използват този подход поради проблеми със сигурността. Те пазят паролите в друг файла, наричан *shadow password fail*, който се намира в */etc/shadow*. Когато паролите са записани в този файл, полето за парола съдържа *x*. Ето как изглежда един shadow файл:

*root:$1$EnI1gPEs$LwH91OIgMRGwPcdqlleU21:12209:0:::::*

*bin:\*:9797:0:::::*

*daemon:\*:9797:0:::::*

*adm:\*:9797:0:::::*

*lp:\*:9797:0:::::*

*sync:\*:9797:0:::::*

*shutdown:\*:9797:0:::::*

*halt:\*:9797:0:::::*

*mail:\*:9797:0:::::*

*news:\*:9797:0:::::*

*uucp:\*:9797:0:::::*

*operator:\*:9797:0:::::*

*games:\*:9797:0:::::*

*ftp:\*:9797:0:::::*

*smmsp:\*:9797:0:::::*

*mysql:\*:9797:0:::::*

*rpc:\*:9797:0:::::*

*gdm:\*:9797:0:::::*

*pop:\*:9797:0:::::*

*nobody:\*:9797:0:::::*

*sshd:\*:9797:0:::::*

*dino:$1$Vd11bYBs$3NV2tT0FK48OR/CYGM/rD0:12209:0:99999:7:::*

*www:$1$m4as.9uy$sOJvGN/fJKeSItUDNxl3e0:12215:0:99999:7:::*

*ffsearch:!:12313:0:99999:7:::*

*stinger:$1$u8M/xs7r$YgtlAmXbGsppdUNk6/onw/:12400:0:99999:7:::*

*postfix:!:12411:0:99999:7:::*

 Когато потребителя въведе паролата си, програмата login прави проста проверка и сравнява двете пароли. Ако те съвпаднат потребителя се допуска до системата, в противен случай достъпа му се отказва.

Въпреки, че преобразуването на паролата до нейния кодиран вариант е кратковременна операция, то обратната операция отнема много време. Полето за парола може да се използва и за ограничаване на достъпа. Това става, като се въведе \* на мястото на кодираната парола. Това се използва за потребителски имена като nobody или lp. Обратно, ако това поле се остави празно, то дадения потребител ще може да се включва в системата без парола.

Всеки потребител има асоцииран уникален номер (UID), като чрез него операционната система идентификацира потребителя. Повечето UNIX системи използват идентификаторите от 0 до 99 за специални UID, а числата от 100 нагоре за потребителите. Slackware използва различна схема, а именно – от числата от 1000 нагоре се използват за потребителите.

И в двата примерни файла има около дузина потребителски имена създадени автоматично от операционната система. Предназначението на по-важните от тях е:

* Потребителя root е суперпотребителя (UID 0)

* demon се използва за системни процеси. Използва се само, за да притежава някои процеси и за да се установяват правилно техните права.

* bin е собственика на изпълнимите файлове

* adm притежава журналните и лог файловете

* uucp се използва за UUCP комуникация

Повечето от тях имат звездичка в полето за парола, което забранява използването им при логин.

Всички потребители са организирани в групи. Всяка група подобно на потребителските имена притежава уникален идентификационен номер (GID). Тя представлява обединение на няколко потребителя по някакви причини. Например в една група могат да бъдат хората от един отдел, а от друга – вашите приятели или клиенти. Така е по-лесно организирането на правата върху определени файлове или други устройства. Всеки потребител може да принадлежи на няколко групи, но в определен момент той е член само на една от тях. Това е свързано с правата на достъп, тъй като Линукс може да присъедини на потребителя само един идентификатор на група.

Информацията за групите се съдържа във файла /etc/group, който има следния вид:

*root::0:root*

*bin::1:root,bin,daemon*

*daemon::2:root,bin,daemon*

*sys::3:root,bin,adm*

*adm::4:root,adm,daemon*

*tty::5:*

*disk::6:root,adm*

*lp::7:lp*

*mem::8:*

*kmem::9:*

*wheel::10:root*

*floppy::11:root*

*mail::12:mail*

*news::13:news*

*uucp::14:uucp*

*man::15:*

*games::20:*

*slocate:x:21:*

*smmsp::25:smmsp*

*mysql::27:*

*rpc:x:32:*

*gdm::42:*

*ftp::50:*

*pop::90:pop*

*nobody::98:nobody*

*nogroup::99:*

*users::100:nobody*

*console:x:101:*

*sshd::33:sshd*

*www:x:102:dino,www*

*ffsearch:x:103:www*

*utmp::22:*

*shadow::43:*

*postdrop:x:104:*

Формата на файла е следния:

*име\_на\_групата:парола:GID:списък\_с\_потребители*

* Името на групата е уникално символно име
* Паролата на групата трябва да бъде въведена от потребителя, ако той пожелае да се присъедини към нея. Не всички дистрибуции използват това поле.
* GID е уникалния групов идентификатор
* Списъкът на потребителите съдържа всички потребители, които принадлежат на тази група

  Всяка Линукс система има няколко системни групи – такива са bin, mail, uucp, sys и др. Присвояването на потребители към тези групи е лоша идея, тъй като това ще предизвика даването на прекалено големи права на тези потребители.

Всяка Линукс дистрибуция притежава вградени програми за добавяне, премахване и промяна на потребителите. Най-простия начин е ръчното редактиране на passwd файла. Това обаче може да предизвика непредвидими ситуации, ако се допусне грешка. В този случай полето за парола се оставя винаги празно, а паролата се добавя след това. При ръчна промяна на passwd и group може да се използват съответно командите *vipw* и *vigr*. Те извикват специална версия на редактора*vi*, която следи промените във файловете да бъдат направени така, че да не се достига до повреда на файловете. Освен това може да се използват командите *adduser* или *useradd* (за добавяне на потребители), *userdel* (за премахване на потребители) и *usermod* (за промяна на детайлите на потребителя).

Повечето дистрибуции притежават и програми за манипулация с групи. Това са *groupadd*,*groupdel* и *groupmod*. Командата groups показва към кои групи принадлежи дадения потребител.

В една добре конфигурирана Линукс система не би трябвало да се работи с root потребителя. В някои случаи обаче се налага от неговото използване. Логичния начин е потребителя да се излезе от системата и след това да влезе като root (ако притежава неговата парола, разбира се). Този подход не е много удобен и затова има алтернатива на него. Това е използването на командата*su.* При извикването й без допълнителни параметри, то тя се опитва да превключи към root потребителя (разбира се ще се наложи да въведете неговата парола). Освен този неявен начин на употреба, командата позволява превключването към който и да е съществуващ потребител:

*#su dino*

*$*

От този момент нататък суперпотребителя работи с привилегиите на потребителя dino. В този случай парола не се изисква, но когато обикновен потребител иска да превключи да друг, система винаги изисква парола. Излизането от този режим става чрез натискането на *Ctrl+D* или чрез командата *exit*.

**Добавяне на потребител**

Силно се препоръчва да създадем потребители за работа в системата, понеже с потребителят root може да създавате, редактирате или изтривате всеки файл в нея. Което означава риск, тъй като това може да доведе до увреждане на системата в случай на допусната грешка от наша страна.

Команди, необходими за създаване на нов потребител **adduser**.

Пример:

# adduser user1

Тази команда създава нов запис във файла /etc/passwd. Този файл съдържа информация, като потребителско име, парола, работна директория и шел по подразбиране.

**Забележка:** Не се препоръчва ръчно редактирането на файла /etc/passwd .

Можем да променим паролата на всеки потребител, ако сме влезли в сесия като root, или нашата собствена парола, в случай, че влезете в своя профил, с помощта на командата **passwd**.

Пример:

# passwd user1

**Добавяне на групи**

GNU/Linux е многопотребителска операционна система, и потребителите могат да влизат в състава на една или няколко групи. Съществуването на групи прави по лесно администрирането на системата, тъй като позволява задаване на правила по групи, вместо да се прави това за всеки потребител.

Има три основни начина да направите това.

* Добавяне на потребител в няколко групи в момента на неговото създаване .
* Добавяне на потребител към група, когато вече съществува .
* Промяна на съществуващ потребител, от една група в друга .

**Добавяне на потребител в няколко групи в момента на неговото създаване**

В примера създаваме потребител ivo, който ще бъде в група users по подразбиране , но също така ще бъде и в група internet .

# useradd -m -g users -G internet ivo

Нека да разгледаме, какво означават опциите които използвах ме в примера .

* -m Ако вече не съществува, създава домашната директория на потребителя.
* -g Име или номера на групата където да бъде добавен (групата трябва да съществува) .
* -G Група или групи в които потребителя също може да бъде добавен (ако са повече от една група, трябва да бъде отделени с запетая , без да се оставя празно място).

**Забележка:** Ако групата internet не съществува то тя трябва да бъде създадена първо .

# groupadd internet

**Забележка:** За да проверите дали една група съществува .

$ egrep -i "^internet" /etc/group

или

$ getent group internet

**Добавяне на потребител към група, когато вече съществува**

Ако потребителя вече съществува и групата също, използваме .

# usermod -a -G internet ivo

**Промяна на съществуващ потребител, от една група в друга**

Да предположим, че не искаме Иво ivo да принадлежи към групата на потребителите users, а в групата за премахнати retired .

usermod -g retired ivo

***Това са някои от най използваните опции , може да погледнете в man useradd , man groupadd , man usermod за повече информация .***

# Директорията / etc / skel

|  |  |
| --- | --- |
|  | Директорията */ etc / skel* съдържа файлове и директории, които автоматично се копират в [*домашна директория*](http://www.linfo.org/home_directory.html) на нов потребител, когато такъв потребител е създаден от програмата *useradd* .  Домашна директория, наричана още *директория* за *влизане* , е директорията на [Unix-подобни](http://www.linfo.org/unix-like.html)[операционни системи,](http://www.linfo.org/operating_systems_list.html) която служи като хранилище за лични файлове, директории и програми на потребителя, включително лични конфигурационни файлове. Също така е и директорията, в която потребителят за първи път [влиза](http://www.linfo.org/login_def.html) след [влизане в](http://www.linfo.org/login_def.html) системата. Директорията */ etc* и нейните поддиректории съдържат много важни конфигурационни файлове за системата.  Програмата *useradd* се намира в директорията */ usr / sbin /* , а на повечето системи е достъпна само за [*коренния*](http://www.linfo.org/root.html) (т.е. административен) потребител. В някои системи тази програма може да се нарича *adduser* .  / etc / skel позволява на системния администратор да създаде домашна директория по подразбиране за всички нови потребители на компютър или мрежа и по този начин да се увери, че всички потребители започват със същите настройки или *среда* .  Няколко потребителски конфигурационни файла са поставени в / etc / skel по подразбиране, когато операционната система е инсталирана. Обикновено те могат да включват *.bash\_profile* , *.bashrc* , *.bash\_logout* , *dircolors* , *.inputrc* и *.vimrc* . Точките, предхождащи имената на тези файлове, показват, че те са [*скрити файлове*](http://www.linfo.org/hidden_file.html) , т.е. файлове, които обикновено не се виждат, за да се избегне визуален хаос и да се намалят шансовете за случайно повреждане.  Съдържанието на / etc / skel може да се види с помощта на *командата ls* (ie, *list* ) с нейната опция *-a* (която показва всички файлове и директории, включително скритите), т.е.  ls -a /etc/skel  Местоположението на / etc / skel може да бъде променено чрез редактиране на реда, който започва с SKEL = в конфигурационния файл */ etc / default / useradd* . По подразбиране този ред казва SKEL = / etc / skel.  Обикновено е по-добре да запазите / etc / skel колкото е възможно по-малко и да поставите конфигурационните елементи на цялата система в глобални конфигурационни файлове като */ etc / profile* . Това е така, защото последното прави много по-лесно актуализирането на файловете на съществуващите потребители, тъй като настройките му влизат в сила веднага след включването на системата и се прилагат за нови потребители и за стари употреби.  Когато даден потребител е премахнат от системата от администратор с командата *userdel* , домашната директория на този потребител, включително файловете и директориите, които са копирани в нея от / etc / skel, остава непокътната.  Името на директорията *skel* се извлича от думата *skeleton* , защото файловете, които тя съдържа, формират основната структура на домашните директории на потребителите. |

# **Разберете малко повече за / etc / profile и / etc / bashrc**

Наскоро работих по проблем, при който приложението не запазваше настройката umask, зададена в профила на root потребителите или /etc/profile. След разглеждане на въпроса малко изглеждаше, че въпросното приложение се прилага само umask настройка, която е била поставена в /etc/bashrcи няма дори да приеме стойности, които са приложения собствени стартови скриптове.

След като направих малко проучени, научих малко повече за това какво точно правят тези файлове, разликите между тях и когато те се изпълняват. Реших, че това би било добро нещо за споделяне, тъй като не е тема, която се появява много често.

## За какво се използва / etc / profile?

Ако сте използвали Linux за известно време, вероятно сте запознати с .profileили .bash\_profileфайлове в домашната директория. Тези файлове се използват за задаване на екологични елементи за потребителска обвивка. Елементи като umask и променливи като PS1или PATH.

Най- /etc/profileфайл не е много по-различна обаче се използва за задаване на цялата система за околната среда променливи за потребители черупки. Променливите понякога са същите, които са в .bash\_profile, но този файл се използва за задаване на начален PATHили PS1за всички потребители на системата.

### /etc/profile.d

В допълнение към настройките за околната среда, /etc/profileще изпълни скриптовете в него /etc/profile.d/\*.sh. Ако планирате да настроите собствените си системни променливи за околната среда, се препоръчва да поставите конфигурацията си в скрипт в рамките на /etc/profile.d.

## За какво се използва / etc / bashrc?

Подобно на .bash\_profileтова, обикновено ще виждате .bashrc файл във вашата домашна директория. Този файл е предназначен за задаване на псевдоними на команди и функции, използвани от потребители на bash shell.

Точно както /etc/profileе и широката версия на системата .bash\_profile. За /etc/bashrcRed Hat и /etc/bash.bashrcUbuntu е широката версия на системата .bashrc.

Интересно е, че при реализацията на Red Hat /etc/bashrcсъщо се изпълняват скриптовете на обвивката в рамките на, /etc/profile.dно само ако потребителската обвивка е интерактивна обвивка (известен още като Shell)

## Кога се използват тези файлове?

Разликата между изпълнението на тези два файла зависи от типа на извършеното влизане. В Линукс можете да имате два типа потребителски обвивки, интерактивни черупки и неинтерактивни черупки. Използва се интерактивна обвивка, при която потребителят може да взаимодейства с обвивката, т.е. Докато неинтерактивна обвивка се използва, когато потребителят не може да взаимодейства с обвивката, т.е. изпълнение на bash скриптове.

Разликата е проста, /etc/profileизпълнява се само за интерактивни черупки и /etc/bashrcсе изпълнява за интерактивни и неинтерактивни черупки. Всъщност в Ubuntu се /etc/profileизвиква /etc/bashrcдиректно.

### Интерактивна обвивка срещу неинтерактивна обвивка

За да се покаже пример за интерактивна обвивка срещу неинтерактивна обвивка, ще добавя променлива в двете /etc/profileи /etc/bash.bashrcв моята Ubuntu система.

#### / И т.н. / профил

*# grep TEST /etc/profile*

export TESTPROFILE=1

#### /etc/bash.bashrc

*# grep TEST /etc/bash.bashrc*

export TESTBASHRC=1

#### Интерактивна обвивка

В по-долу пример показва интерактивна черупка, в този случай двете /etc/profileAND /etc/bash.bashrcе екзекутиран.

*# su -*

*# env | grep TEST*

TESTBASHRC=1

TESTPROFILE=1

#### Неинтерактивна обвивка

В този пример изпълняваме команда чрез SSH, която не е интерактивна; защото това е неинтерактивна обвивка, само /etc/bash.bashrcфайлът се изпълнява.

*# ssh localhost "env | grep TEST"*

root@localhost's password:

TESTBASHRC=1

## заключение

В моя случай децата на приложенията не разпознават зададената стойност на umask, /etc/profileно разпознават стойността в /etc/bashrc. Това ми казва, че подпроцесът започва като неинтерактивна обвивка. Докато предложеният начин за модифициране на променливи на околната среда е да се добави шел скрипт /etc/profile.dв моя случай е по-добре да се зададе umask стойност в /etc/bashrc.