2. Команден интерпретатор  
• Роля на командния интерпретатор, видове;  
• Login, logout, виртуални терминали, графичен login, мрежови login;  
• Изпълнение на команди и навигация в конзолатa;  
• Системна информация – who, whoami, w, finger, users, groups, uname;  
• Root концепция, смяна на потребител su / sudo , UID, GID, id, tty ;  
• Помощ – help, man, info;  
• Среда и променливи на средата;  
• Хронология на командите – history.

Част 2

**Команден интерпретатор**

• Роля на командния интерпретатор, видове;  
• Login, logout, виртуални терминали, графичен login, мрежови login;  
• Изпълнение на команди и навигация в конзолатa;  
• Системна информация – who, whoami, w, finger, users, groups, uname;  
• Root концепция, смяна на потребител su / sudo , UID, GID, id, tty ;  
• Помощ – help, man, info;  
• Среда и променливи на средата;  
• Хронология на командите – history.

В тази част ще се запознаем какво е терминал и за какво се ползва както и видовете терминали според предназначението . Също и как се влиза и излиза от линукс ще се научим също и да навигираме в конзолата както и още няколко важни команди и още една част от концепцията на линукс. Работа със хелп файлове и потребители и роли .

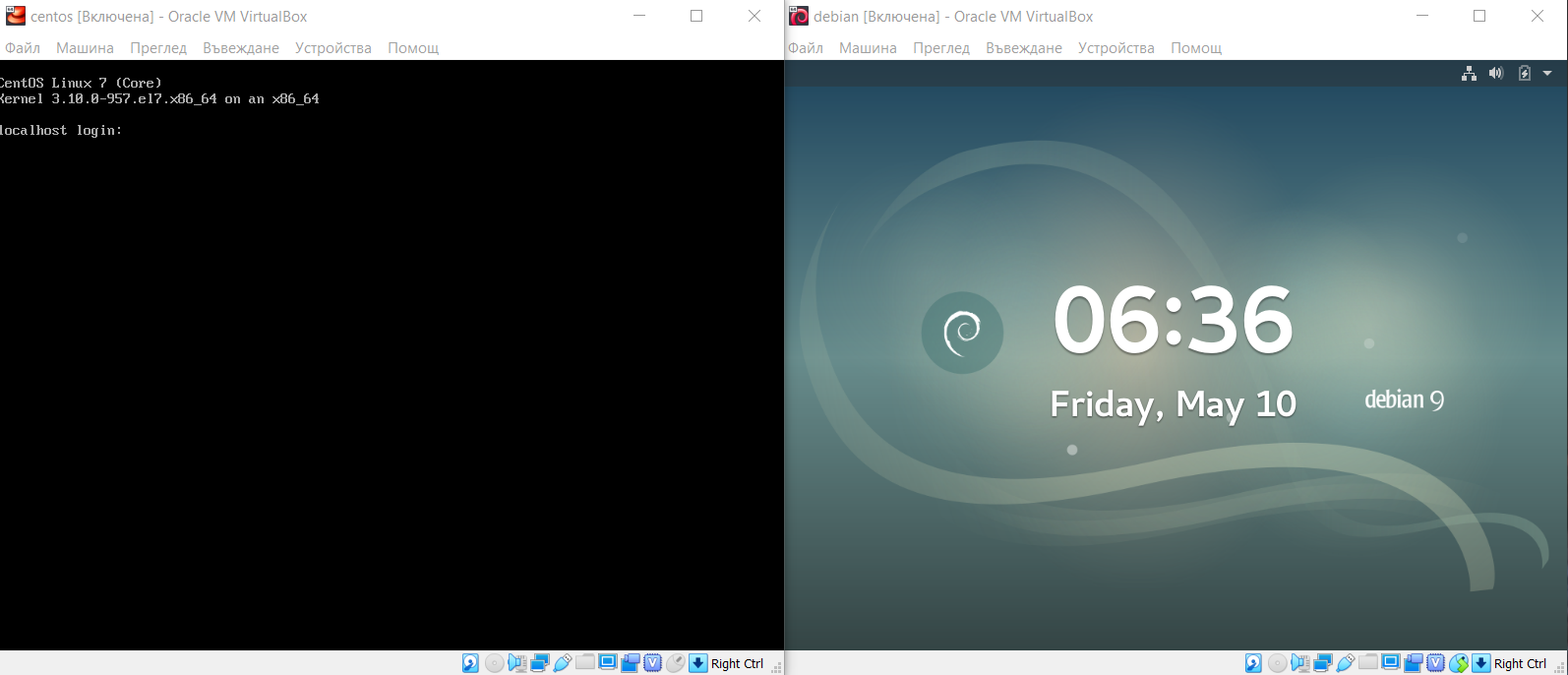
Така че да започваме .

Командният интерпретатор (на английски: Shell, шел – обвивка) е потребителски интерфейс за достъп до услуги на съответната операционна система. Като цяло обвивките (shell) на операционните системи използват или интерфейс с команден ред или графичен потребителски интерфейс.

Shell е програма, която приема команди от потребителя, и стартира програмите асоциирани с тези команди, или с други думи това е софтуерен компонент, който осигурява интерфейс за потребителите.

Терминът много често се отнася до обвивката (shell) на операционната система, т.е. външния слой на интерфейса между потребителя и вътрешността на операционната система (ядрото). Шелът осигурява достъп до услуги на ядрото на операционната система.

Операционните системи обикновено имат два вида обвивки: от команден ред и графична. Командният интерпретатор от команден ред осигурява интерфейс с команден ред (CLI) на операционната система, а графичният команден интерпретатор осигурява графичен потребителски интерфейс (GUI). И при двете категории основната цел на шела или командния интерпретатор е да извика или да „стартира“ друга програма; въпреки това, шеловете често имат и допълнителни възможности, като например показване на съдържанието на директории и файлове.

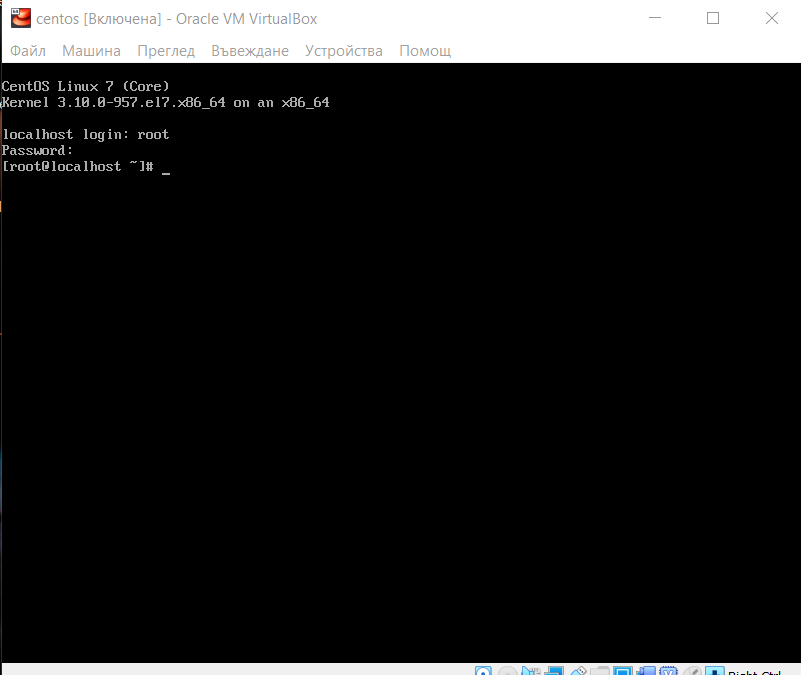
От предишната секция както забелязахме центос и дебиан след инсталацията изглеждат различно тоест имат различни шелове 

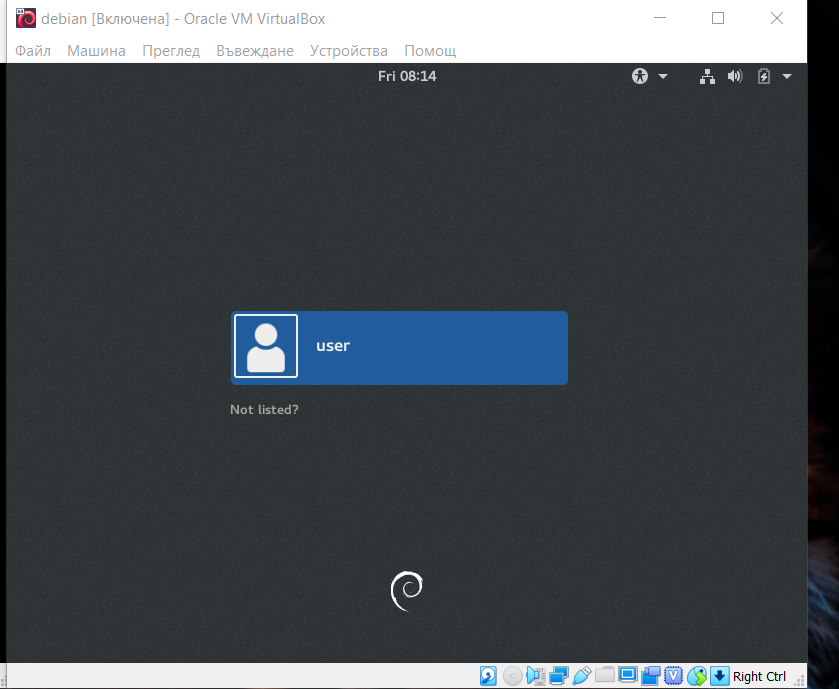
Ще разгледаме работата във тях както и основна навигация в операционната система както и различни видове шелове .

Но първо нека да влезем в нашата система както е видно от снимката в Centos пита за потребителско име и парола

Ще влезем с root и паролата която зададохме по време на инсталацията

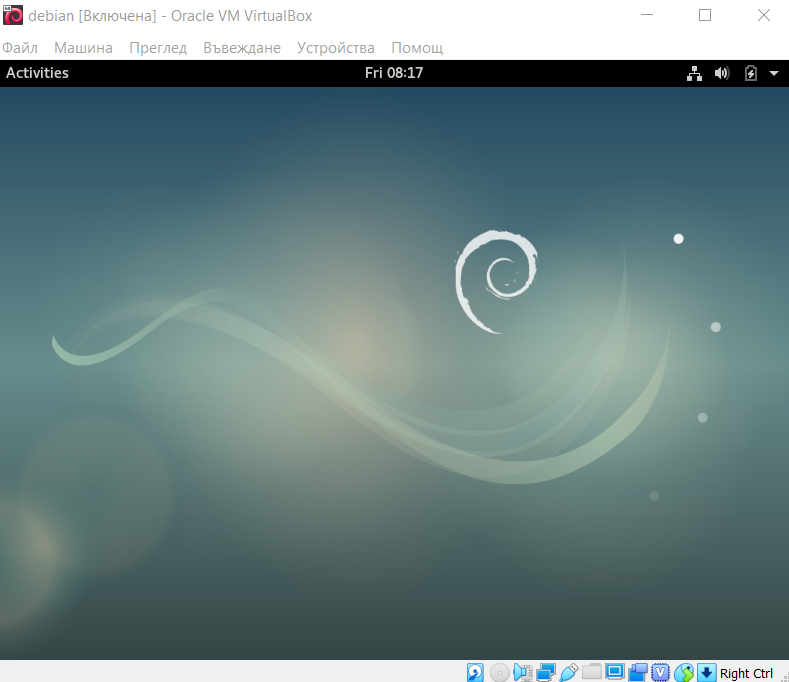
Ще забележите че като се въвежда паролата полето седи празно



В дебиан е малко по различно 

Както забелязвате на екрана не е root а е потребителят който си създадохме по време на инсталацията

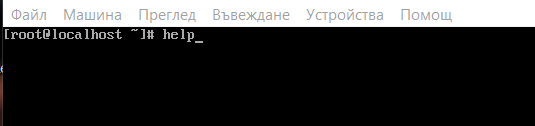
И паролата за него



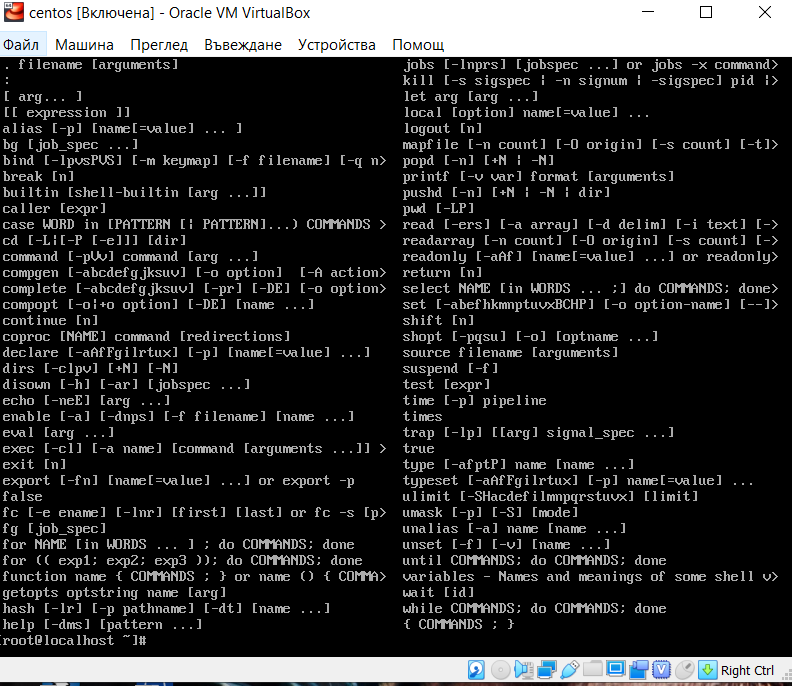
Както е видно от снимките двете операционни системи зареждат различни шелове в случаят центос команден шел а дебиан графичен такъв.

Сега да се запознаем с операционната система по отблизо за целта ще видиме хелп секцията

В прозорчето на центос напишете help и натиснете ентър

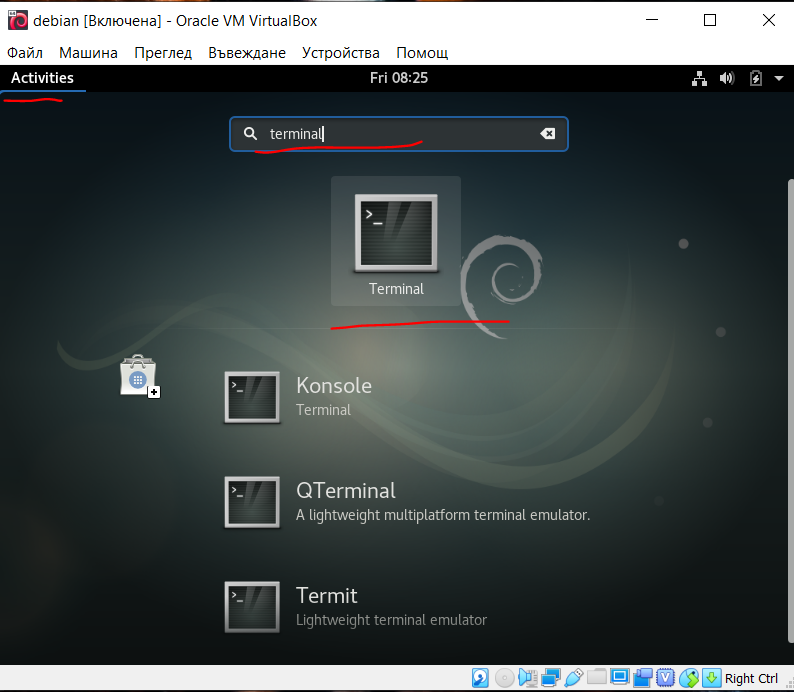


Появява се следният прозорец

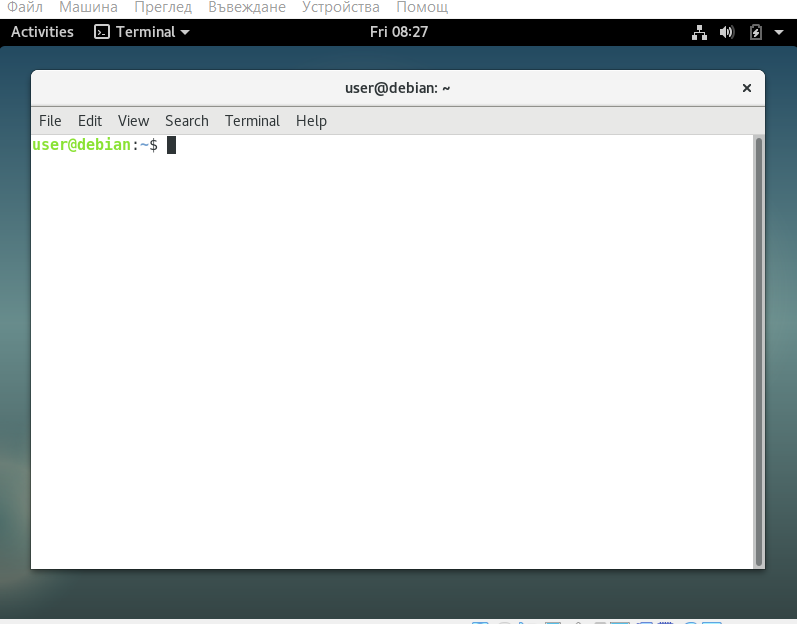


Ще направим същото и в Дебиан

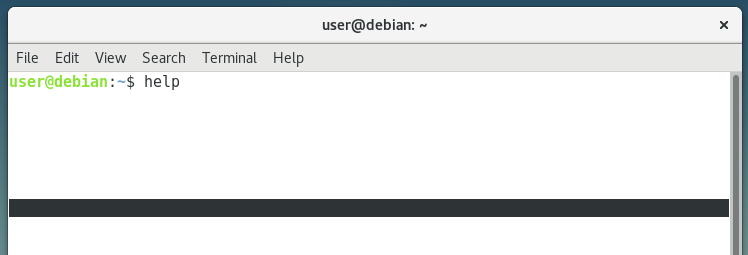
За целта горе в дясно където пише activities и в поленцето за търсене въвеждаме terminal



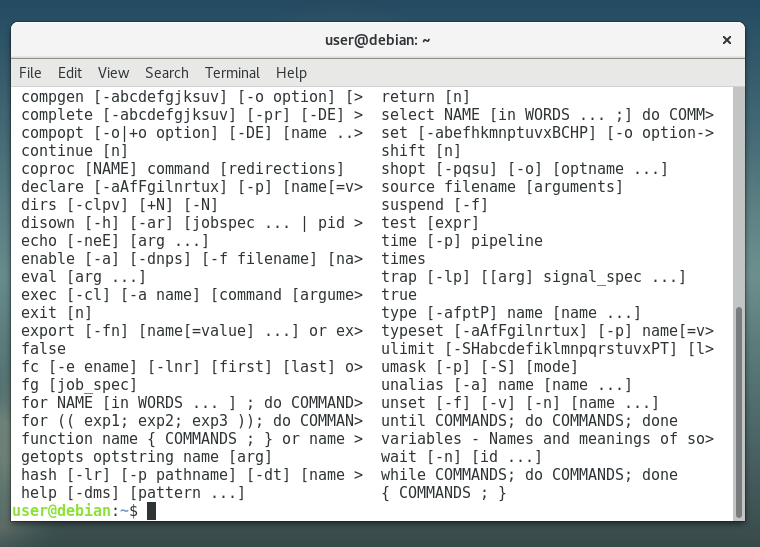
След като г остартираме се появява терминалът



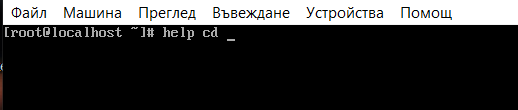
Въвеждаме help и натискаме enter



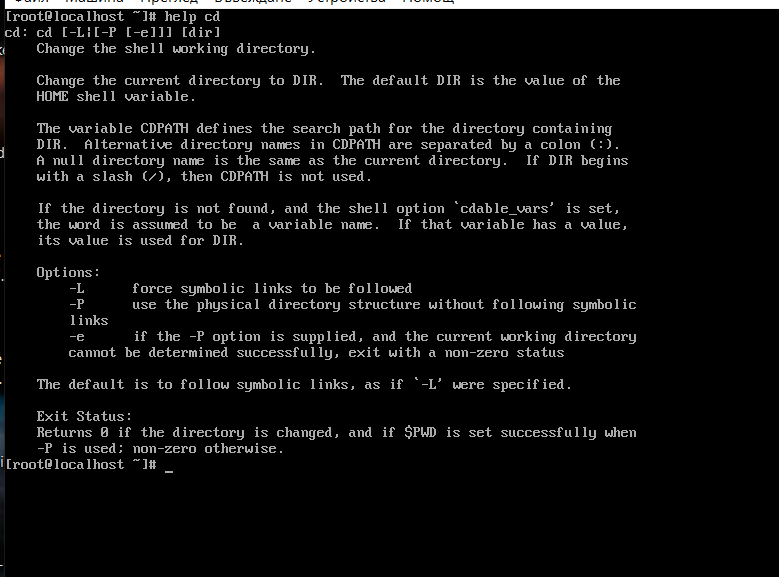
Както се вижда излизат по голямата част от командите които може да ползваме



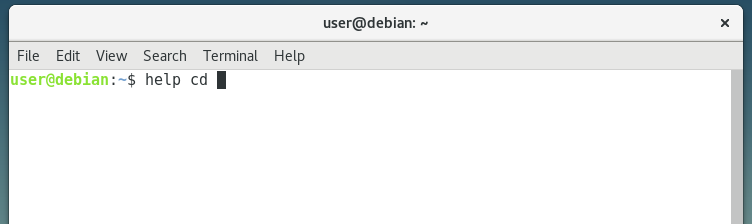
Следва да разберем малко повече информация за командите за това ще използваме хелп за специфична команда да кажем за класическото cd



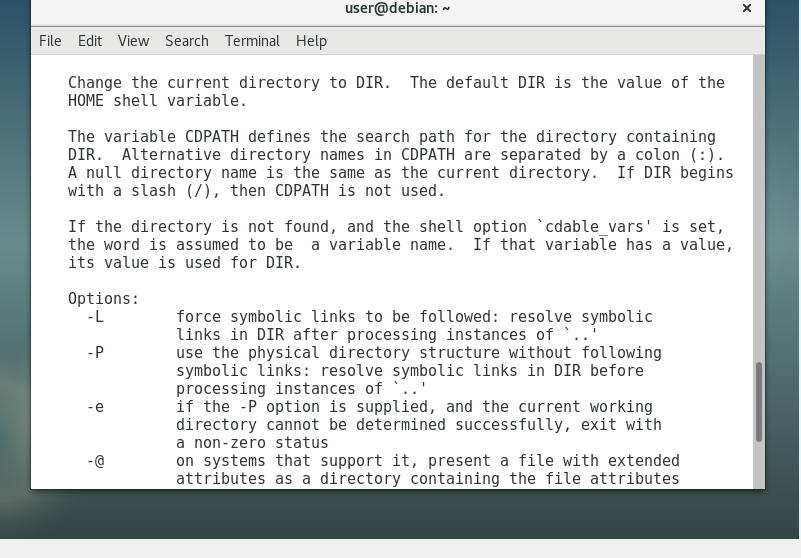
Както се вижда хелпа ни дава пълна информация за cd командата



В аналогичен начин ще постъпиме и в дебиан



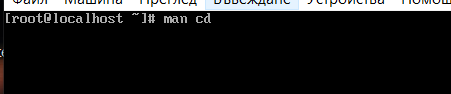
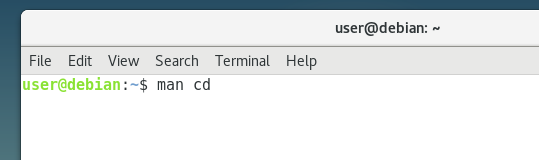
И резултатът



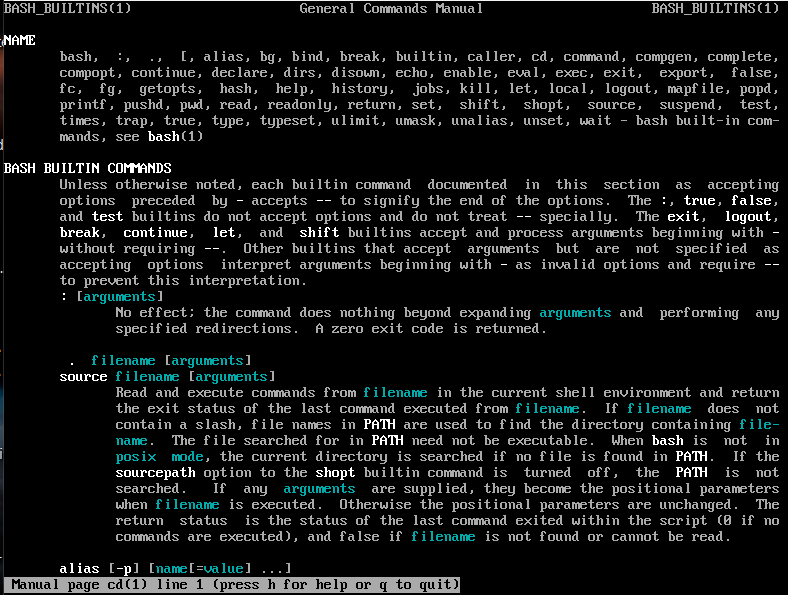
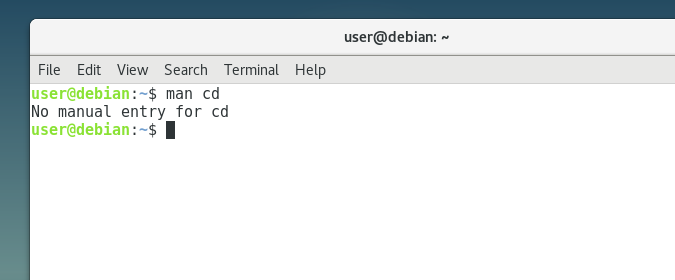
Друг полезен начин да разбереме всичко за командите е man или ръководството на разработчика за въпросната команда

В центос и дебиан ще изпълниме следната команда

man cd

И съответно резултатът

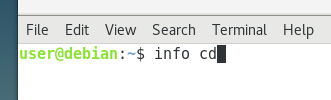
 

Видно е ве дебиан не предоставят информация за командата cd докато в центос дават доста подробна информация за да продължиме да четем на екрана просто натискаме ентър а ако искаме да излезем натискаме q

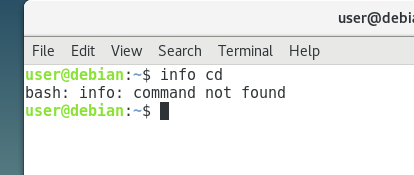


Друга полезна команда е info

Ще изпълниме в нашите виртуални машини следната команда info cd

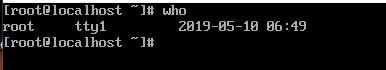
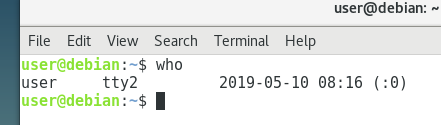
 

Както се вижда центос връща доста информация докато дебиан казва че командата не е намерена

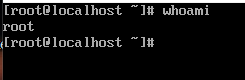
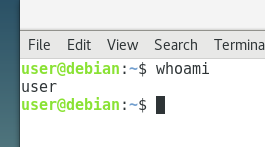
 

Сега да разбереме малко повече подробности за нас в системата за целта ще използваме следният набор от команди who, whoami, w, finger, users, groups, uname;

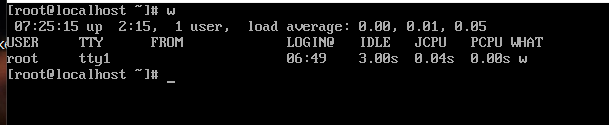
Ако ги изпълниме в терминала всяка команда ще ни даде различна информация например who

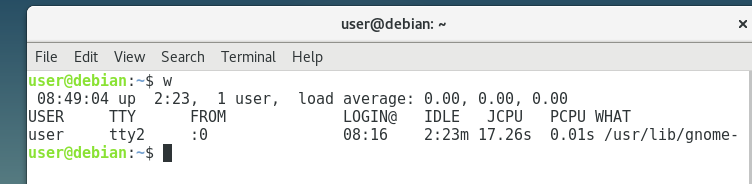
 

Аналогично и за whoami

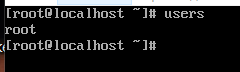
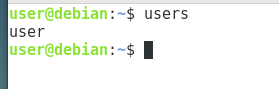
Командата w ни дава малко по детайлна информация която ще я разгледаме след малко



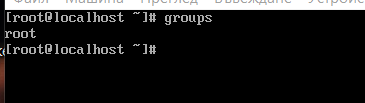
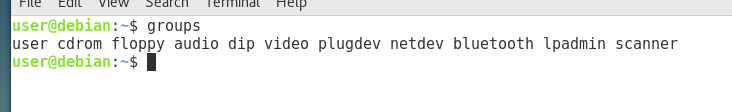


Както се вижда двете операционни системи връщат различни данни

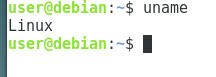
Командата users ни дава списък със всичко влезли в машината потребители

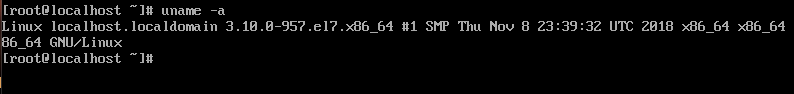
Командата groups ни дава информация потребителят на кои групи е член

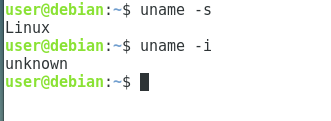
 

Командата uname ни дава основна информация за системата която ползваме

Обаче като всяка команда може да ис е добавят различни параметри и съответно да изкара различни резултати





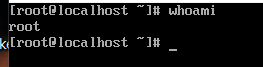
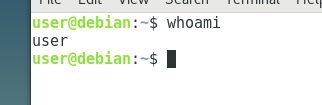
Повече за командата може да разберем с man uname или на сайтът <https://linux.die.net/man/1/uname>

Същото се отнася и за предишните команди ако не сте сигурни какво търсите като информация винаги проверявайте в man pages или info.

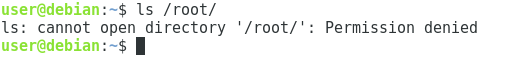
Всички промени в системата се правят от тъй нареченият root потребител който има абсолютните права да променя всичко във операционната система .

За целта трябва да станеме root потребител

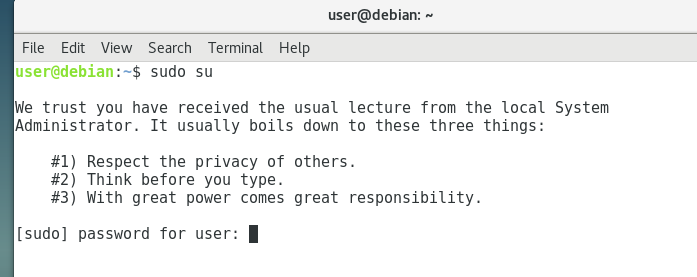
Проверяваме кои сме в системата ( аз предпочитам с whoami тъй като дава информацията която ми трябва )

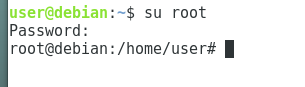
 

Както е видно в центос вече сме влезнали с root потребителят докато в дебиан сме с обикновен потребител който има ограничени права за да ви го демонстрирам ще изпълня следната команда ls /root която единственото коеот прави е да покаже съдържанието на папката /root ( повече за ls по късно).

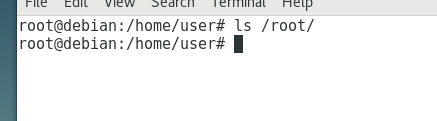
Както се вижда от снимката получаваме отказани права в дебиан . За целта ще влезем като root и там

Команда е sudo su и съответно нашата парола за потребителят

Ако в дебиан видите следният екран  или съобщение че командата не е намерена има и друг начин комдата е su ( съкратено от switch user ) за целта искаме да станем root за това изпълняваме следната команда su root 

Само че тук вече не е паролата за нашият потребител ами root паролата която сме задали по време на инсталацията

Както се вижда вече и в дебиан сме root ( по аналогичен начин седът нещата и в centos ) . да се върнеме на предишната команда ls /root

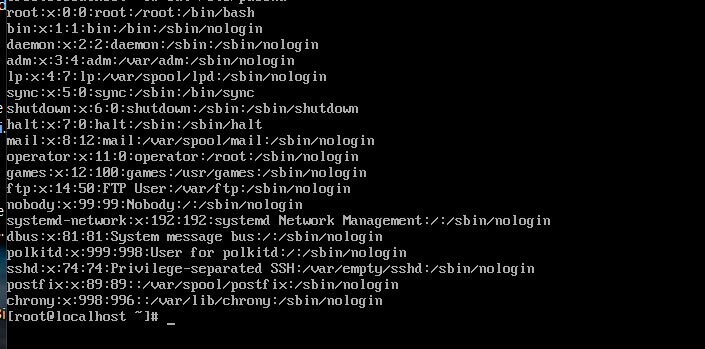
 

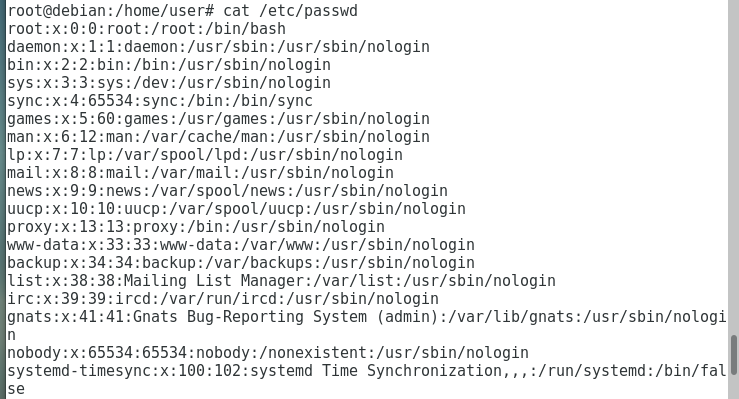
Както се вижда вече и в дебиан сме супер потребители ако искаме са смениме потребителят тоест пак да се върнем на user изпълняваме su user както виждате в случаят не ни пита за парола тъй като вече сме я въвели веднъж следователно я знаем обаче ако изпълниме пак su root отново ще пита за root паролата .

Потребителите в линукс операционните системи имат ИД номера по подобие на ЕГН номера . Това се ползва с цел да няма два еднакви потребителя в системата

Като първите 999 са резервирани за системните потребители

Ето как изглеждат въпросните потребители и потребителски групи в центос и дебиан



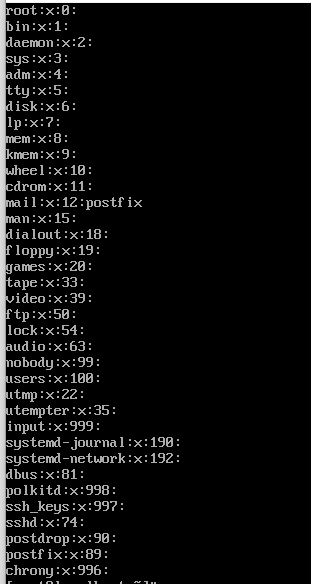
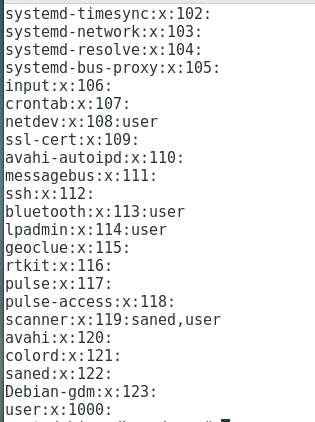




Както се вижда от горните скройншоти root потребителят е с id 0 което се води с най висок приоритет а нашият потребител е с id 1000

Освен потребителските ИД номера има и групови които не е задължително да съвпадат с потребителското например да вземем потребителят adm както се вижда той има потребителско ид 3 обаче груповото му е 4

Това се дължи на фактът че не е задължително всички потребители да са членове на своята си група а доста често са членове на няколко групи както ще видите от следващият скрийншот

Както е видно в дадена група може да има повече от един потребител .

Концепцията е много проста по лесно се управляват групи отколкото индивидуални потребители .

Променливите на средата в операционните системи, базирани на ядрото на Linux, са променливи, които съдържат текстова информация, използвана от други програми по време на стартиране. Обикновено те включват общи системни параметри на графичната и командната обвивка, данни за потребителските настройки, местоположението на определени файлове и много други. Стойностите на такива променливи се посочват, например, чрез числа, символи, пътеки към директории или файлове. Благодарение на това много приложения бързо получават достъп до определени настройки и също така става възможно потребителят да промени или създаде нови опции.

## Работа с променливи на средата в Linux

В тази статия бихме искали да засегнем основната и най-полезната информация, която се отнася до променливите на околната среда. Освен това ще демонстрираме начини да ги преглеждате, променяте, създавате и изтривате. Познаването на основните опции ще помогне на начинаещите потребители да се ориентират в управлението на такива инструменти и да се справят с тяхната стойност в дистрибуциите на операционната система. Преди да започнем анализа на най-важните параметри, бих искал да говоря за тяхното разделяне на класове. Такова групиране се определя, както следва:

1. Системни променливи Тези опции се зареждат незабавно при стартиране на операционната система, съхраняват се в определени конфигурационни файлове (те ще бъдат разгледани по-долу) и са достъпни за всички потребители и за цялата операционна система като цяло. Обикновено тези параметри се считат за най-важни и често използвани при пускането на различни приложения.
2. Потребителски променливи. Всеки потребител има своя собствена домашна директория, където се съхраняват всички важни обекти, включително конфигурационните файлове на потребителските променливи. От тяхното име вече е ясно, че те се използват за конкретен потребител в момента, когато той е упълномощен чрез местния "Терминал" . Те работят на отдалечена връзка.
3. Локални променливи. Има параметри, които се прилагат само в една сесия. Когато бъде завършена, те ще бъдат окончателно изтрити и за да стартирате отново всичко ще трябва да бъде създадено ръчно. Те не се записват в отделни файлове, а се създават, редактират и изтриват с помощта на съответните конзолни команди.

### **Конфигурационни файлове за потребителски и системни променливи**

Както вече знаете от описанието по-горе, два от трите класа Linux променливи се съхраняват в отделни файлове, където се събират общи конфигурации и разширени параметри. Всеки такъв обект се зарежда само при подходящи условия и се използва за различни цели. Отделно, бих искал да подчертая следните елементи:

* /ETC/PROFILE - един от системните файлове. На разположение на всички потребители и на цялата система, дори и с дистанционно влизане. Единственото ограничение за това е, че параметрите не се приемат при отваряне на стандартния "Терминал" , т.е. в това местоположение, няма да работят стойности от тази конфигурация.
* /ETC/ENVIRONMENT - по-широк аналог на предишната конфигурация. Той работи на системно ниво, има същите опции като предишния файл, но сега без никакви ограничения дори с отдалечена връзка.
* /ETC/BASH.BASHRC - файлът е само за локална употреба, няма да функционира по време на отдалечена сесия или връзка през Интернет. Той се изпълнява за всеки потребител поотделно при създаването на нова терминална сесия.
* .BASHRC - отнася се за конкретен потребител, се съхранява в домашната си директория и се изпълнява всеки път, когато се стартира нов терминал.
* .BASH\_PROFILE е същото като .BASHRC , само за отдалечаване , например, когато се използва SSH.

### **Списък на главните системни и потребителски променливи**

Благодарение на горните инструкции вече знаете как бързо можете да определите всички текущи параметри и техните стойности. Остава само да се справим с основните. Искам да обърна внимание на следните въпроси:

* DE . Пълното име е Desktop Environment. Съдържа името на текущата среда на десктопа. Операционните системи на ядрото на Linux използват различни графични черупки, така че е важно приложенията да разберат коя е в момента активна. Това е мястото, където променливата DE помага. Пример за неговите стойности е gnome , mint , kde и т.н.
* PATH - определя списъка с директории, в които се търсят различни изпълними файлове. Например, когато се изпълнява една от командите за търсене и достъп до обекти, те получават достъп до тези папки, за да търсят бързо и да прехвърлят изпълними файлове с посочените аргументи.
* SHELL - запазва опцията за активна командна обвивка. Такива черупки позволяват на потребителя самостоятелно да регистрира определени скриптове и да изпълнява различни процеси, използвайки синтаксиси. Най-популярната обвивка е bash . Списък с други общи команди за запознаване може да се намери в нашата друга статия на следния линк.

Вижте също: [Често използвани команди в терминала на Linux](https://soringpcrepair.com/main-linux-commands-in-terminal/" \t "_blank)

* HOME - всичко е достатъчно просто. Този параметър указва пътя до началната папка на активния потребител. Всеки потребител има различен и изглежда по следния начин: / home / user . Обяснението на тази стойност също е лесно - тази променлива, например, се използва от програмите за установяване на стандартното местоположение на техните файлове. Разбира се, все още има много примери, но това е достатъчно за запознаване.
* BROWSER - съдържа команда за отваряне на уеб браузър. Тази променлива най-често определя браузъра по подразбиране и всички други помощни програми и софтуер имат достъп до тази информация, за да отварят нови раздели.
* PWD и OLDPWD . Всички действия от конзолата или графичната обвивка идват от определено място в системата. Първият параметър е отговорен за текущата находка, а вторият показва предишния. Съответно техните стойности се променят доста често и се съхраняват както в потребителските конфигурации, така и в системните.
* TERM . Има голям брой терминални програми за Linux. Споменатата променлива съхранява информация за името на активната конзола.
* RANDOM - съдържа скрипт, който генерира случайно число всеки път от 0 до 32767 при достъп до тази променлива. Тази опция позволява на друг софтуер да работи без свой собствен генератор на случайни числа.
* EDITOR - отговаря за отварянето на редактора на текстови файлове. Например, по подразбиране можете да намерите пътя / usr / bin / nano там , но нищо не ви пречи да го промените на друго. За по-сложни действия с тест, VISUAL е отговорен и стартира, например, редактора vi .
* HOSTNAME е името на компютъра, а USER е името на текущия профил.

### **Изпълнение на команди с новата променлива на обкръжението**

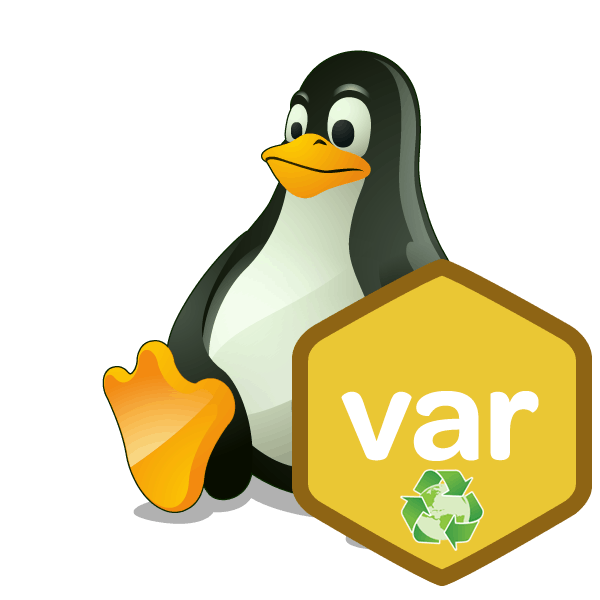
Можете да промените опцията на всеки параметър за известно време, за да стартирате определена програма с нея или да извършите други действия. В този случай в конзолата трябва само да регистрирате env VAR=VALUE , където VAR е името на променливата, а VALUE е неговата стойност, например пътя до папката /home/user/Download .

Следващият път, когато видите всички параметри чрез командата printenv спомената по-горе, ще видите, че зададената стойност е променена. Въпреки това, тя ще стане така, както е била по подразбиране, веднага след следващия достъп до нея, а също така функционира само в активния терминал.

### **Задаване и изтриване на променливи на локалната среда**

От материала по-горе вече знаете, че локалните параметри не са запаметени във файлове и са активни само в рамките на текущата сесия, а след неговото завършване се изтриват. Ако желаете сами да създадете и изтриете такива опции, трябва да направите следното:

# **Променливи на Linux среда**



Променливите на средата в операционните системи, базирани на ядрото на Linux, са променливи, които съдържат текстова информация, използвана от други програми по време на стартиране. Обикновено те включват общи системни параметри на графичната и командната обвивка, данни за потребителските настройки, местоположението на определени файлове и много други. Стойностите на такива променливи се посочват, например, чрез числа, символи, пътеки към директории или файлове. Благодарение на това много приложения бързо получават достъп до определени настройки и също така става възможно потребителят да промени или създаде нови опции.

Съдържанието

* [Работа с променливи на средата в Linux](https://soringpcrepair.com/linux-environment-variables/#___Linux)
  + [Конфигурационни файлове за потребителски и системни променливи](https://soringpcrepair.com/linux-environment-variables/#i)
  + [Вижте списък с променливи на системната среда](https://soringpcrepair.com/linux-environment-variables/#i-2)
  + [Списък на главните системни и потребителски променливи](https://soringpcrepair.com/linux-environment-variables/#i-3)
  + [Изпълнение на команди с новата променлива на обкръжението](https://soringpcrepair.com/linux-environment-variables/#i-4)
  + [Задаване и изтриване на променливи на локалната среда](https://soringpcrepair.com/linux-environment-variables/#i-5)
  + [Добавете и премахнете потребителски променливи](https://soringpcrepair.com/linux-environment-variables/#i-6)
  + [Създаване и изтриване на променливи на системната среда](https://soringpcrepair.com/linux-environment-variables/#i-7)

## Работа с променливи на средата в Linux

В тази статия бихме искали да засегнем основната и най-полезната информация, която се отнася до променливите на околната среда. Освен това ще демонстрираме начини да ги преглеждате, променяте, създавате и изтривате. Познаването на основните опции ще помогне на начинаещите потребители да се ориентират в управлението на такива инструменти и да се справят с тяхната стойност в дистрибуциите на операционната система. Преди да започнем анализа на най-важните параметри, бих искал да говоря за тяхното разделяне на класове. Такова групиране се определя, както следва:

1. Системни променливи Тези опции се зареждат незабавно при стартиране на операционната система, съхраняват се в определени конфигурационни файлове (те ще бъдат разгледани по-долу) и са достъпни за всички потребители и за цялата операционна система като цяло. Обикновено тези параметри се считат за най-важни и често използвани при пускането на различни приложения.
2. Потребителски променливи. Всеки потребител има своя собствена домашна директория, където се съхраняват всички важни обекти, включително конфигурационните файлове на потребителските променливи. От тяхното име вече е ясно, че те се използват за конкретен потребител в момента, когато той е упълномощен чрез местния "Терминал" . Те работят на отдалечена връзка.
3. Локални променливи. Има параметри, които се прилагат само в една сесия. Когато бъде завършена, те ще бъдат окончателно изтрити и за да стартирате отново всичко ще трябва да бъде създадено ръчно. Те не се записват в отделни файлове, а се създават, редактират и изтриват с помощта на съответните конзолни команди.

### **Конфигурационни файлове за потребителски и системни променливи**

Както вече знаете от описанието по-горе, два от трите класа Linux променливи се съхраняват в отделни файлове, където се събират общи конфигурации и разширени параметри. Всеки такъв обект се зарежда само при подходящи условия и се използва за различни цели. Отделно, бих искал да подчертая следните елементи:

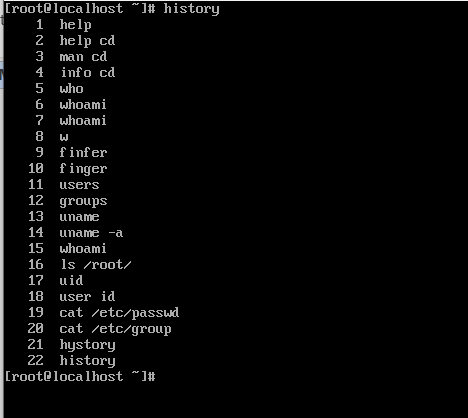
* /ETC/PROFILE - един от системните файлове. На разположение на всички потребители и на цялата система, дори и с дистанционно влизане. Единственото ограничение за това е, че параметрите не се приемат при отваряне на стандартния "Терминал" , т.е. в това местоположение, няма да работят стойности от тази конфигурация.
* /ETC/ENVIRONMENT - по-широк аналог на предишната конфигурация. Той работи на системно ниво, има същите опции като предишния файл, но сега без никакви ограничения дори с отдалечена връзка.
* /ETC/BASH.BASHRC - файлът е само за локална употреба, няма да функционира по време на отдалечена сесия или връзка през Интернет. Той се изпълнява за всеки потребител поотделно при създаването на нова терминална сесия.
* .BASHRC - отнася се за конкретен потребител, се съхранява в домашната си директория и се изпълнява всеки път, когато се стартира нов терминал.
* .BASH\_PROFILE е същото като .BASHRC , само за отдалечаване , например, когато се използва SSH.

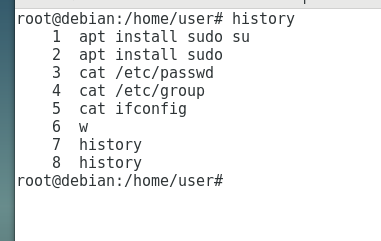
Като във всяка операционна система и в линукс се пази история на командите които са изпълнявани или тъй наречената команда **history**

При изпълнение на командата ще видиме на екрана списък със всички команди които се изпълнили .

Ще попитате защо е важно това , просто е доста често в ежедневната работа се налага да се изпълняват едни и същи команди както ще разберете по нататък някои от които са доста дълги за да бъдат изписвани отново и отново . Така че да се запознаем със командата и какво може .

Това е проста демонстрация какво излиза като отговор когато се изпълни командата в конзолата на линукс





Това което прави впечатление е че срещу всяка команда има съответно номер . Това е с цел лесно да се извика повторно командата например ако искаме да изпълним отново командата history не е необходимо да я изписваме цялата може просто да я извикаме по следният начин !22 в нашата центос конзола удивителната означава че викаме повторно команда и съответно посочваме нейният номер . в дебиан е по същият начин .

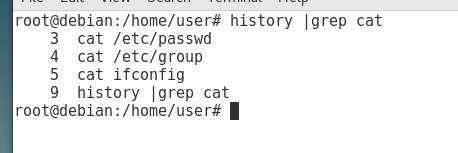
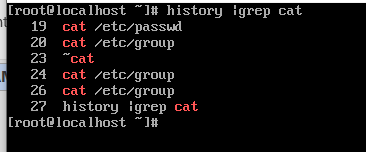
Другият начин е със изписване на част от командата

Например в центос (важи и за всички останали линукси ) да кажем че искаме отново да извикаме списъка със всички потребители (командата е cat /etc/passwd) но за да не го пишеме отново цялото в конзолата е достатъчно да сложим удивителна и cat тоест !cat това ще изпълни последната команда от списъка където сме извикали съответната команда само че в случаят извиква списъка със групите .

Също така може и да търсиме в съответният списък с команди . Командата за търсене на определена команда е със следният синтаксис

history | grep cat

и съответно резултатите



Както се вижда излизат само редовете където се съдържа съответната команда cat и може да извикаме тази която ни трябва с ! и номерът и . Другият начин е да търсиме рекурсивно тоест във всичко за целта в конзолата изпълняваме клавишната комбинация от контрол+r (cntr+r)

Както ще забележите шелът се смени и излезна поле

 в момента в който започнеме да въвеждаме някакъв текст ще започнат да се показват всички команди в които се съдържа когато сме готови просто натискаме ентър и командата се изпълнява отново.

Може също и да променяме команди със субституция ( тоест заместваме една команда с друга ) например нашата cat команда със ls например

Синтаксисът е ^cat^ls^

Имайте предвид че има команди които не винаги са съвместими и може да излезе съобщение за грешка като в този случай



Другото коеот може да правим е да трием част от историята или цялата история

Ако желаем да изтрием само определен ред например 3

Командата ще е history –d 3 това ще изтрие само ред 3

Ако желаем да изтрием цялата история командата е history –c ( в случаят променливата d е delete а c clear)