# 1 Тема 5. Теоретическая часть

*Рассматривая базовые концепции ОС* [5, подраздел 1.7], было отмечено, что *кон- цепция пользователя* является второй по значимости после *концепции файла*.

*С другой стороны*, *концепция пользователя* интенсивно используется и в *концеп- ции процесса*, что связывает их воедино, образуя три базовых концепции ОС.

*В зависимости от контекста*, термин пользователя понимается как:

* *человек*, работающий за терминалом компьютера;
* *концептуальная основа* выделения ***владельцев***, ***группы*** и ***других***, связанные с концепциями файла и процесса;
* *система обозначений*, использующая ***имена*** и ***числовые идентификаторы***, распространяемая на все ПО ЭВМ;
* *система разграничений*, ***ограничивающая*** и ***специализирующая права*** ис- пользования ПО ЭВМ.

*Кроме перечисленных выше*, имеются специальные режимы использования ОС и ЭВМ, которые связаны с *настройкой, инсталляцией и промежуточными этапами загрузки ОС*.

*Подобное многообразие взглядов* порождает еще большее многообразие зависи- мостей, что влияет не только на работу отдельной ОС, но и распространяясь на со- циум, порождает новые технологические подходы к архитектуре современных ОС.

*Чтобы успешно разобраться* в указанных проблемах и освоить данную тему, учеб- ный материал разбит на четыре части:

* *многопользовательский режим* работы ОС, как альтернатива однопользова- тельскому, порождающий многобразие взглядов, уже указанных выше;
* *разграничение прав пользователей*, как техническая основа функционирова- ния ОС;
* *login и система PAM*, как реализация эффективного использования ПО ОС;
* *команды управления пользователями*, как управляющая часть организации многопользовательского режима работы ОС.
  1. **Однопользовательский и многопользовательский режимы**

**работы ОС**

*Первоначально*, когда ОС еще не было, а использовалась пакетная обработка прог- рамм или программы супервизоры, понятие пользователя использовалось в чисто внешнем организационном плане: *как лицо*, передающее программу в службу ВЦ (вычислительного центра) на исполнение.

*В 1974 году*, Гари Килделл преподаватель информатики в аспирантуре военно-мор- ского колледжа в Монтерее (Калифорния) закончил писать свою ОС для микро- компьютера intel 8080, которую он назвал CP/M.

**CP/M** - Control Program for Microcomputers.

*В 1980 году*, фирма Microsoft приобрела у Seattle Computer лицензию на довольно сырую и недоработанную операционную систему 86-DOS. Немного доработав, Билл Гейтс предложил использовать ее в качестве MS-DOS в первом персональном компьютере фирме IBM.

**MS-DOS** — доработанная 86-DOS фирмы Seattle Computer, способная запускать все программы ОС CP/M.

*Характерная черта* MS-DOS и CP/M — *однозадачный режим работы на изолиро- ванном компьютере*, что собственно говоря и не требовало ставить вопрос об ис- пользовании их многими пользователями. Эта черта в последующем отразилась и на ОС MS Windows:

* *автоматическое монтирование* всех файловых систем при старте ОС;
* *отсутствие концепции пользователя* в низкоуровневых структурах ФС.

*История ОС UNIX* начинается с середины 60-х годов, на фоне проекта операцион- ной системы MULTICS, который разрабатывался в Bell Labs., подразделении гиган- та AT&T.

*Первое издание UNIX*, появившееся в ноябре 1971 года, работало на PDP-11/20 без MMU и аппаратной защиты памяти. Стабильность ее работы и устойчивость к сбо- ям была не на высоте. *Мультипрограммности тоже не было*, но пути к файлам уже появились. Была документация к таким системным вызовам:

break, cemt, chdir, chmod, chown, close, creat, exec, exit, fork, fstat, getuid,gtty, ilgins, intr, link, mkdir, mount,open, quit, read, rele, seek, setuid, smdate, stat, stime, stty, tell, time, umount, unlink,wait, write.

Из языков программирования поддерживались ассемблер, B, BASIC, FORTRAN.

Языка С еще не было. Хотя явное упоминание о поддержке многопользовательско- го режима отсутствует, наличие команд *chown*, *getuid* и *setuid* говорит о том, что *в файловой системе концепция пользователя уже была заложена*.

*Современное понятие однопользовательского режима означает* не тот факт, что ОС не может поддерживать многопользователский режим, а то что:

* *или отключен контроль* разграничения прав пользователей, при одновремен- ной изляции ЭВМ от внешних воздействий, например, отключение от сети;
* *или остановлена работа* программ всех пользователей, кроме администра- тора, например, суперпользователя ***root***.

*Например*, современные ОС загружаются в два этапа:

* *на первом этапе*, после загрузки и запуска ядра ОС специальным загруз- чиком, например GRUB, ядро распаковывает в оперативную память *времен- ную файловую систему* и запускает первый процесс ***init***;
* сам процесс ***init*** — *обычно скрипт*, выполняемый интерпретатором shell, ус- танавливает необходимые *модули* ОС, ищет и монтирует *корневую* файловую систему, создает *терминальные устройства* и запускает на них программы

***login***, удаляет временную файловую систему и ***завершает работу***; все это делается **в** *однопользовательском режиме ОС*;

* *на втором этапе*, пользователи, которые начинают проходить процедуру

***login***, работают уже *в многопользовательском режиме ОС*.

**Замечание**

Практически всегда, под *именем пользователя* понимается *контекст*, соответствую-

щий понятию *владелец*, который относятся применительно к файлам и процессам.

**Информационное обеспечение** многопользовательского режима ОС, прежде всего, поддерживается группой системных файлов.

*Файл* ***/etc/passwd***, каждая строка которого имеет формат:

username:password:UID:GID:GEOS:homedir:shell

где

|  |  |
| --- | --- |
| username | Имя пользователя, используемое для входа в систему. Содержит слово (ранее - до 8 букв). Заглавные буквы не допускаются. |
| password | Hash-код пароля. Сейчас ставится символ **x**, а hash-код пароля перене- сен в файл ***/etc/shadow***. |
| UID | Число-идентификатор пользователя. |
| GID | Число-идентификатор основной группы, в которую входит пользова- тель. |
| GEOS | Любая информация. |
| homedir | Домашняя директория пользователя. |
| shell | Командный интерпретатор пользователя, который запускается при его входе в систему. Список возможных интерпретаторов находится в файле **/etc/shells**. Если имя пользователя не предназначено для интерактивной работы с ОС, то указывается **/bin/false** или ***/bin/nologin*** |

*Учитывая большую важность этой информации*, содержимое файла ***/etc/passwd***

дублируется в файл ***/etc/passwd-***.

*Информация о группах пользователей* и дубль этой информации хранятся в файлах

**/etc/group** и **/etc/group-**, в формате:

**groupname:password:GID:userlist**

где

|  |  |
| --- | --- |
| groupname | Имя группы с теми же ограничениями, что и имена пользователей. |
| password | Hash-код пароля (если пароль имеется). Сейчас ставится символ **x**, а hash-код пароля перенесен в файл ***/etc/gshadow***. |
| GID | Число-идентификатор группы. |
| userlist | Список пользователей, входящих в группу, разделенных запятыми. Первый пользователь в списке — администратор группы. |

* 1. **Разграничение прав пользователей**

*Общая парадигма концепции пользователя* подразумевает, что *все пользователи ОС* ***работают автономно*** *и не мешают друг другу*, кроме системного администрато- ра.

Это достигается двумя основными мерами:

* *каждый пользователь* имеет право работать только с теми файлами, директо- риями и файловыми системами, к которым он имеет доступ;
* *пользователь* **root**, с идентификатором ***UID=0***, может делать абсолютно все.

*На самом деле*, имя пользователя имеет вспомогательное второстепенное значение. *Главным показателем пользователя* является его идентификатор ***UID***: с увели- чением номера ***UID*** права пользователя уменьшаются.

*Аналогичный критерий* справедлив для групп пользователей, права которых опре- деляются идентификатором ***GID***.

*Условно*, все пользователи разделяются на две категории:

* *системные пользователи* — ***root***, ***sysadm*** и другие администраторы;
* *обычные пользователи* — те, которые используют прикладное программное обеспечение ОС и не занимаются администрированием.

*Условность такого разделения* подтверждается тем фактом, что в первых ОС иден- тификаторы обычных пользователей начинались с номера ***100***.

*Со временем*, разработчики прикладного ПО стали столь интенсивно использовать идентификаторы, что было принято решение:

* *системные* пользователи — ***UID < 999***;
* *пользователь live*-дистрибутива — ***UID=999*** и ***GID=999***;
* *обычные* пользователи — ***UID > 999***.

**Замечание** Хотя часто, *при создании нового пользователя*, обычно создается и группа с таким же именем и ***GID=UID***, - это не является обязательным требованием:

* при создании, новый пользователь может быть сразу включен в любую, уже существующую группу;
* любой пользователь, без ограничений, может быть включен в произвольное число групп.

*Как правило*, обычному пользователю доступны:

* *все файлы и каталоги его домашней директории*, положение которой задано системной переменной ***HOME***;
* *права записи в каталоги* ***/tmp*** и ***/var/tmp***;
* *права монтирования и демонтирования* внешних устройств, которые пропи- саны в файле ***/etc/fstab*** с опцией ***user***.

***Временная смена прав доступа*** на права другого пользователя достигается ко- мандой:

**su [-] [username]**

при этом, ему придется ***набрать пароль*** того пользователя, под чьим именем он собирается работать:

* *Если присутствует первый аргумент* команды ***su***, то *произойдет смена до- машней директории* и выполнятся скрипты входа типа ***~/.profile***.
* Если второй аргумент не указан, то подразумевается пользователь ***root***.

**Выполнене команд от имени пользователя *root***, для обычного пользователя, выполняется командой:

**sudo список\_команд;**

при этом, ему придется набрать *свой собственный пароль*.

**Замечание**

Если пользователю необходимо уточнить под чьими именами и идентификаторами он работает, можно воспользоваться командами ***whoami*** и ***id***.

**Смена прав пользователя ОС** связана с его действиями в системе.

**Действия пользователя в системе** определяется работой программ (процессов), которые пользователь запускает. Чтобы определить, с какими правами работает процесс, вводятся дополнительные понятия, связанные с *реальными и эффектив- ными идетификаторами пользователя*.

**Действительные** *(реальные) ID пользователя и ID группы* — это числовые (двух- байтовые) значения ***UID*** и ***GID***, записанные в файлах ***/etc/passwd*** и ***/etc/group*** во время создания пользователя в системе.

**Эффективные** *ID пользователя и ID группы* — это числовые (двухбайтовые) значе- ния, которые учитываются в системе при выполнении конкретного процесса:

* *дочерний процесс*, создаваемый системным вызовом *fork(…)*, получает эф- фективные ***ID*** от своего родителя;
* *процесс*, модифицируемый одним из системных вызовов *exec(…)*, устанавли- вает эффективные ***ID*** в зависимости от значений битов ***SUID*** и ***SGID***, присут-

ствующих в поле *i\_mode* индексного десткриптора файла (см. [8, подраздел 1.5, таблицы 1.9 и 1.10]): если биты ***SUID*** и ***SGID*** — установлены, то эффективные ***ID*** берутся из дескриптора файла, а если нет, то устанавливают- ся в значения действительных (реальных) ***ID*** пользователя, запустившего процесс.

**Сохраненные** *ID пользователя и ID группы* — это числовые (двухбайтовые) значе- ния *первоначальных эффективных ID*, которые сохранены в памяти процесса с помощью системных вызовов *getuid(...)* и *getgid(...)*, сразу же после завершения системного вызова *exec(...)*.

**Замечание** Установка новых значений эффектиных ID выполняется с помощью системных вызовов *setuid(...)* и *setgid(...)*.

На рисунке 1.1, представлен вывод эффектиных идентификаторов процесса shell, запущенного пользователем upk в окне терминала.

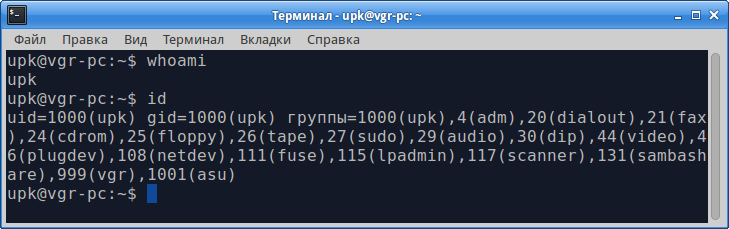


Рисунок 1.1 — Вывод эффективных ID пользователя upk

## Login и система доступа Linux-PAM

**Любому пользователю**, для нормальной работы с ОС, необходимо пройти *проце- дуру регистрации в системе*: получить имена, идентификаторы, пароли и место для работы. Указанную процедуру выполняет администратор ОС.

*В результате регистрации*:

* *имена, идентификаторы и пароли* будут записаны в соответствующие фай- лы: ***/etc/passwd***, ***/etc/group***, ***/etc/shadow*** и ***/etc/gshadow***;
* *в директории /home* будет создана директория с именем пользователя;
* *в директорию /home* будут перенесены директории и файлы, находящиеся в

*директории /etc/skel*, которые составляют начальный скелет рабочей области

любого пользователя.

**После загрузки ОС** и перехода ее в многопользовательский режим, запускается процесс «*Манеджер сеанса*», который контролирует и обеспечивает процедуры входа пользователя в систему:

* *в случае положительного завершения* процедры входа в систему, дополни- тельно контролируется содержимое директории /home/$USER и, случае необ-

ходимости, в нее добавляются нужные файлы или модифицируются старые, а также *устанавливаются эффективные идентификаторы* для процессов сан- са пользователя;

* *в случае негативного завершения* процедры входа в систему или *негативного контроля* содержимого директории /home/$USER, выполняются дополни-

тельные процедуры, которые заканчиваются перезапуском «***Менеджера се-***

***ансов***».

**Замечание**

Среда исполнения, о которой говорилось ранее, представляет собой *одну системную среду исполнения* и *пользовательские среды исполнения*, по одной на каждый вход в систему.

**Процедура входа в ОС** может быть: *текстовой*, когда на консоль терминала выво- дится приглашение *login:*, или *графической*, когда выводится некоторое стилизо- ванное окно приглашения.

*В любом случае*, требуется набрать имя и пароль, а возможно и другие сведения, например, домен или язык работы с системой. Это зависит от настроек «***Менедже- ра сеансов***».

*После ввода необходимой информации* начинается с *процедуры* ***login***, которая под- разделяется на:

* *идентификацию (аутентификацию)*, подразумевающую совпадение имени и пароля, зарегистрированных в системе;
* *авторизацию*, подразумевающую создание среды для работы программ поль- зователя и фиксирование прав, которыми пользователь обладает.

*Фактически*, авторизация не заканчивается завершением утилиты ***login***. Она прово- дится постоянно, когда пользователь обращается к файлам или взаимодействует с процессами.

**Поскольку методы авторизации могут быть различны**, а пользователю даже приходится обращаться к программам, требующим смены пользователя, то *смена парадигмы обеспечения безопасности работы ОС*, приводит к перезаписи большо- го количества системного ПО.

*В 1995 году*, **OSF** (***Open Software Foundation***) — фонд открытого программного обеспечения - приступил к разработке *системы PAM* (***Pluggable Authentication Modules***) — *заменяемые модули идентификации*.

**Замечание**

**К середине 90-х годов**, проблема безопасности ОС стала столь критичной, что постави- ла под сомнение архитектурные принципы различных систем.

**Система PAM** разрабатывалась для UNIX-подобных систем.

**MS Windows** имеет свою оригинальную систему защиты.

**Система PAM** введена *для создания дополнительного уровня защиты* между при- ложениями и различными протоколами и способами идентификации и авториза- ции.

*Модули PAM* — это динамически загружаемые библиотеки, которые находятся в директориях **/lib/security** или **/usr/lib/security**.

*Все приложения* используют универсальный интерфейс, ***PAM API***, а уже модули PAM выбирают стратегию поведения и протоколы согласно файлам конфигурации:

### /etc/pam.conf либо /etc/pam.d/....

**Замечание** Модульная система PAM не столько обеспечивает новый уровень защиты, сколько раз- деляет прикладную часть процессов от части, обеспечиващей защиту их функциони- рования, одновременно централизуя ПО защиты, обеспечивая мобильность его модифи- кации и ускоряя внедрение новых технологий.

**Технологическая концепция** модулей PAM предполагает разделение их на четыре типа:

|  |  |
| --- | --- |
| auth | Выполняют аутентификацию, то есть подтверждают, что пользователь является именно тем, кем он представился в системе. |
| account | Разрешают или запрещают конкретному пользователю вход в систему. Это решение может зависеть от даты, времени суток, системных ресур- сов и т.д. |
| session | Осуществляют действия, которые должны быть выполнены до или после входа пользователя: занесение информации в журнальные файлы, монти- рование устройств и другое. |
| passwd | Изменяют пароль пользователя. |

*Практическая реализация этой концепции* предполагает централизованное исполь- зование файлов конфигурации.

*Сейчас*, файлы конфигурации Linux-PAM находятся в директории */etc/pam.d*, в которой находятся файлы, как правило совпадающие с именами файлов приложе- ний. Например, для программы *sudo*, файл конфигурации: */etc/pam.d/sudo*.

*Каждый файл конфигурации* состоит из отдельных строк, содержащих поля:

**тип\_модуля управляющий\_флаг имя\_модуля аргументы**

|  |  |
| --- | --- |
| тип\_модуля | Один из четырех типов: ***auth, account, session, passwd***. |
| управляющий\_флаг | Флаг, контролирующий поведение PAM в случае успешного или безуспешного результата работы модуля:   * *required* - успешное завершение работы модуля необходимо для успеха всего запроса. Об ошибочном   завершении не будет сообщено до окончания работы всех модулей.   * *requisite* — успешное завершение работы модуля необходимо для успеха всего запроса. Ошибочное   завершение приводит к немедленному возврату управления приложению.   * *sufficient* — в случае успешного завершения, управ- ление немедленно передается приложению. Ошибоч-   ное завершение модуля не учитывается.   * *optional* — результат работы этого модуля не учи- тывается. |
| имя\_модуля | Имя файла, которое должно быть указано с полным путем к |

нему.

аргументы Командная строка, передаваемая модулю.

*Таким образом*, система PAM позволяет разрабатывать систему безопасности без переделки самих приложений.

**Замечание** Чтобы узнать, *какие библиотеки* PAM использует приложение, *лучше* воспользоваться командой (утилитой) **ldd**. Например, для приложения ***su***, имеем:

$ ldd /bin/su

linux-vdso.so.1 (0x00007ffca7a6f000)

libutil.so.1 => /usr/lib/libutil.so.1 (0x00007f28917e5000) libsudo\_util.so.0 => /usr/lib/sudo/libsudo\_util.so.0 (0x00007f28915d3000) libc.so.6 => /usr/lib/libc.so.6 (0x00007f289121c000) libdl.so.2 => /usr/lib/libdl.so.2 (0x00007f2891018000)

/lib64/ld-linux-x86-64.so.2 => /usr/lib64/ld-linux-x86-64.so.2 (0x00007f2891c0a000)

$

## Команды управления пользователями

*Теоретическая концепция* пользователя поддерживается соответствующей инфор- мационной и управляющей инфраструктурой ОС.

*Хотя работа с пользователями* предполагает всего три действия: *создание*, *удале- ние* и *модификацию*, - наличие множества конфигурационных файлов, привязанных к древовидной структуре ФС, превращает сам процесс управления в непростую задачу.

*Чтобы упростить этот процесс*, используются три команды (утилиты): *useradd*, *userdel* и *usermod*, расположенные обычно в директории */usr/sbin*.

**Замечание** Прежде чем выполнять любую из этих команд, следует воспользоваться руководством ***man***, а после — предварительно запустить команды с ключем --***help***.

Общий синтаксис команды *создания нового пользователя* имеет вид:

**useradd [ -A { DEFAULT | method [ ,... ] } ]**

[ -c comment ] [ -d home\_dir ]

[ -e expire\_date ] [ -f inactive\_time ] [ -g initial\_group ]

[ -G group [ ,... ] ]

[ -m [ -k skeleton\_dir ] | -M ] [ -s shell ]

[ -u uid [ -o ]] имя\_пользователя [ -r ]

[ -n ]

Команда требует одного обязательного аргумента — *имя\_пользователя*. Многие другие аргументы команды интуитивно понятны и не требуют пояснения.

**Замечание**

Наличие множества параметров команды создания пользователя требует хорошего навыка и знания как пользователь работает.

Каждый администратор вырабатывает свои правила работы с пользователями. После ввода, команда может задать вопросы в интерактивном режиме.

**Удаление** *пользователя* выполняется командой:

**userdell имя\_пользователя;**

Если пользователь с таким именем существует и, при этом, не находится в системе, то *userdell*:

* *удаляет его домашний каталог* со всеми подкаталогами;
* *удаляет все записи* об этом пользователе из файлов */etc/passwd*, */etc/shadow*, */ etc/group*;
* *возможно, оставляет* временные забытые файлы в директории */tmp*.

**Изменение** *параметров пользователя* выполняется командой *usermod*, в которой большинство аргументов совпадают с аргументами команды *useradd*, но ориенти- рованы на изменение соответствующих параметров.

**Особое место** в управлении пользователями занимает утилита *passwd*, которая уп- равляет паролями пользователя и ограничивает его работу на уровне сеанса.

*Общие правила* применения утилиты passwd:

* *администратор ОС* может изменить пароль любого пользователя;
* *работа утилиты*, практически всегда происходит в интерактивном режиме; например, новый пароль вводится дважды;
* *обычный пользователь* может сменить только свой пароль, предварительно набрав старый.

*Общий синтаксис* команды:

**passwd [параметры] [LOGIN]**

**Замечание**

Обязательно следует изучить параметры команды с помощью руководства ***man passwd***

и запуска ее с ключем —***help***.

*Утилита passwd* имеет множество опций, поэтому рассмотрим наиболее типичные варианты ее применения.

**passwd [ -f | -s ] [ имя ]**

|  |  |
| --- | --- |
| -f | Позволяет изменить поле GEOS в файле ***/etc/passwd***. |
| -s | Позволяет изменить интерпретатор ***shell***, вызываемый при входе пользователя в систему. |

**passwd [ -g ] [ -r | -R ] группа**

|  |  |
| --- | --- |
| -g | Переключает ***passwd*** в режим работы с паролями групп. |
| -r | Удаляет групповой пароль. |
| -R | Закрывает доступ к группе для всех пользователей. |

**passwd [ -x max ] [ -n min ] [ -w warn ] [ -i inact ] имя**

|  |  |
| --- | --- |
| -x max | Максимальное число дней, в течение которых пароль действителен. 9999 — пароль действителен всегда. |
| -n min | В течение скольких дней пользователь не может изменять свой пароль. 0 — может всегда. |
| -w warn | За сколько дней до истечения срока ***max*** пользователю начнут выдаваться предупреждения о необходимости смены пароля. |
| -i inact | Число дней, свыше ***max***, когда пользователь может сменить пароль, иначе он будет заблокирован до вмешательства администратора. |

**passwd { -l | -u | -d | -S } имя**

|  |  |
| --- | --- |
| -l | Временно запретить доступ пользователя в систему. |
| -u | Восстановить доступ пользователя в систему. |
| -d | Удаление пароля пользователя с разрешением входа в систему (без пароля). |
| -S | Получить информацию о пароле пользователя. Например,  $ passwd -S  vgr P 12/21/2012 0 99999 7 -1  $ |