

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

**Кафедра автоматизированных систем управления (АСУ)**

## **ОПЕРАЦИОННЫЕ СИСТЕМЫ**

### **Тема 5. Управление пользователями ОС**

#### **Учебно-методическое пособие**

для студентов уровня основной образовательной программы: **бакалавриат**  
направление подготовки: **09.03.01 - Информатика и вычислительная техника**  
направление подготовки: **09.03.03 - Прикладная информатика**

Разработчик  
доцент кафедры АСУ

**В.Г. Резник**

2021

**Резник В.Г.**

Операционные системы. Тема 5. Управление пользователями ОС. Учебно-методическое пособие. – Томск, ТУСУР, 2021. – 19 с.

Учебно-методическое пособие предназначено для изучения теоретической части и выполнения лабораторной работы №5 по теме «Управление пользователями ОС» учебной дисциплины «Операционные системы» для студентов кафедры АСУ ТУСУР уровня основной образовательной программы бакалавриат направлений подготовки: «09.03.01 - Информатика и вычислительная техника» и «09.03.03 - Прикладная информатика».

## Оглавление

<b>Введение.....</b>	<b>4</b>
<b>1 Тема 5. Управление пользователями ОС.....</b>	<b>5</b>
1.1 Однопользовательский и многопользовательский режимы работы ОС.....	5
1.2 Разграничение прав пользователей.....	8
1.3 Login и система доступа Linux-PAM.....	10
1.4 Команды управления пользователями.....	13
<b>2 Лабораторная работа №5.....</b>	<b>16</b>
2.1 Инфраструктура управления пользователями.....	16
2.2 Реальные и эффективные права пользователя.....	16
2.3 Инфраструктура PAM.....	17
2.4 Команды управления пользователями.....	17
<b>Список использованных источников.....</b>	<b>19</b>

## Введение

Данная тема нашей дисциплины посвящена управлению пользователями ОС. Важность этой темы может показаться не столь очевидной, например, по сравнению с темами управления файловыми системами и процессами, но такое впечатление является обманчивым:

- атрибуты пользователя, связывая файлы и процессы, существенно влияют как на возможность самих вычислений, так и часто на сам результат;
- пользователь как участник процесса функционирования ПО ОС требует выделения изолированной рабочей области, гарантирующей ему сохранность его личных ресурсов, но одновременную возможность использования общих ресурсов;
- пользователь, как участник совместных действий с другими пользователями, становится ответственным за результаты своего функционирования, во всех многообразиях его проявления.

Перечень изучаемых в данной теме вопросов и их место в учебном материале дисциплины «Операционные системы» изложен в источнике [1], основным учебником является [2], а дополнительным [3]. В качестве источника практических задачи используем учебный материал, изложенный в [4]. При необходимости, используются ссылки на материал предыдущих тем, изложенный в [5-8].

Первый раздел, озаглавленный «Тема 5. Теоретическая часть», собственно и содержит описание всех заявленных вопросов. Здесь с разных сторон рассмотрено понятие пользователя ОС, его место в операционной среде исполнения и связи с концепцией файловой системой хранения информации. Дается краткая классификация пользователей и рассматриваются вопросы безопасности их совместной работы. Демонстрационные примеры теоретической части ложатся в основу лабораторной работы по данной теме.

Второй раздел, озаглавленный «Лабораторная работа №5», содержит методический материал по практическому закреплению полученных знаний. Средой исполнения этих работ является ОС УПК АСУ, установленная в учебных классах кафедры АСУ или на личных компьютерах студентов. Успешно выполненной считается работа описанная в личном отчете студента и проверенная преподавателем.

# 1 Тема 5. Управление пользователями ОС

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

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

*В зависимости от контекста*, термин **пользователь** 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* использования 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 ещё большее многообразие зависимостей, что влияет не только на работу отдельной ОС, но и распространяется на социум и порождает новые подходы в архитектуре современных ОС.

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

- 1) *многопользовательский режим* работы ОС, как альтернатива однопользовательскому, порождающий многообразие взглядов, уже указанных выше;
- 2) *разграничение прав пользователей*, как техническая основа функционирования ОС;
- 3) *login и система РАМ*, как реализация эффективного использования ПО ОС;
- 4) *команды управления пользователями*, как управляющая часть организации многопользовательского режима работы ОС.

## 1.1 Однопользовательский и многопользовательский режимы работы ОС

*Первоначально*, когда ОС ещё не было, а использовалась пакетная обработка программ 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 военноморского колледжа в Монтерее (Калифорния) закончил писать свою ОС для микрокомпьютера Intel 8080, названную **CP/M** (*Control Program for Microcomputers*).

*В 1980 году*, фирма **Microsoft** приобрела у **Seattle Computer** лицензию на до-

вольно сырую и недоработанную операционную систему 86-DOS, которую Билл Гейтс предложил фирме IBM использовать в качестве MS-DOS в ее первом персональном компьютере IBM PC.

**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 изолированном компьютере*. Эти черты MS-DOS отразилась и на первых 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 операционной системы **MULTICS**, который разрабатывался в **Bell Labs.**, подразделении гиганта **AT&T**.

*Одно из первых изданий ОС UNIX*, появившееся в ноябре 1971 года, работало на PDP-11/20 без **MMU** и аппаратной защиты памяти. Стабильность ее работы и устойчивость к сбоям 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, FORTRAN, Bi и BASIC. Языка C ещё не было. Хотя явное упоминание о поддержке многопользовательского 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 многопользовательский режим, а то что:

- *или отключён контроль* разграничения прав пользователей, при одновременной изоляции ЭВМ от внешних воздействий, например, отключение от сети;
- *или остановлена работа* программ всех пользователей, кроме администратора, например, суперпользователя **root**.

**Современные ОС** загружаются в два этапа:

- *на первом этапе*, после загрузки и запуска ядра ОС специальным загрузчиком, например GRUB, ядро распаковывает в оперативную память *временную файловую систему* и запускает первый процесс **init**;
- сам процесс **init** — *обычно скрипт*, выполняемый интерпретатором shell, устанавливает необходимые модули ОС, ищет и монтирует *корневую файловую систему*, создаёт *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*, соответствующий понятию *владелец*, который относится 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

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

*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:user list
```

где

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

## 1.2 Разграничение прав пользователей

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

Это достигается двумя основными мерами:

- *каждый пользователь* имеет право работать только с теми файлами, директориями 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 увеличением номера **UID** права пользователя уменьшаются.

*Аналогичный критерий* справедлив для групп пользователей, права которых определяются 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 идентификаторы 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 создаётся и группа с таким же именем и **GID=UID**, - это не является обязательным требованием:

- при создании, новый пользователь может быть сразу включён в любую, уже существующую группу;
- любой пользователь, без ограничений, может быть включён 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 демонтажа* внешних устройств, которые прописаны в файле **/etc/fstab** с опцией **user**.

*Временная смена прав доступа* на права другого пользователя достигается командой:



## su [-] [username]

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

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

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

## sudo список\_команд;

при этом, ему придётся 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 эффективными идентификаторами пользователя*.

**Действительные (реальные) ID пользователя и ID группы** — это числовые (двухбайтовые) 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) значения, которые учитываются в системе при выполнении конкретного процесса:

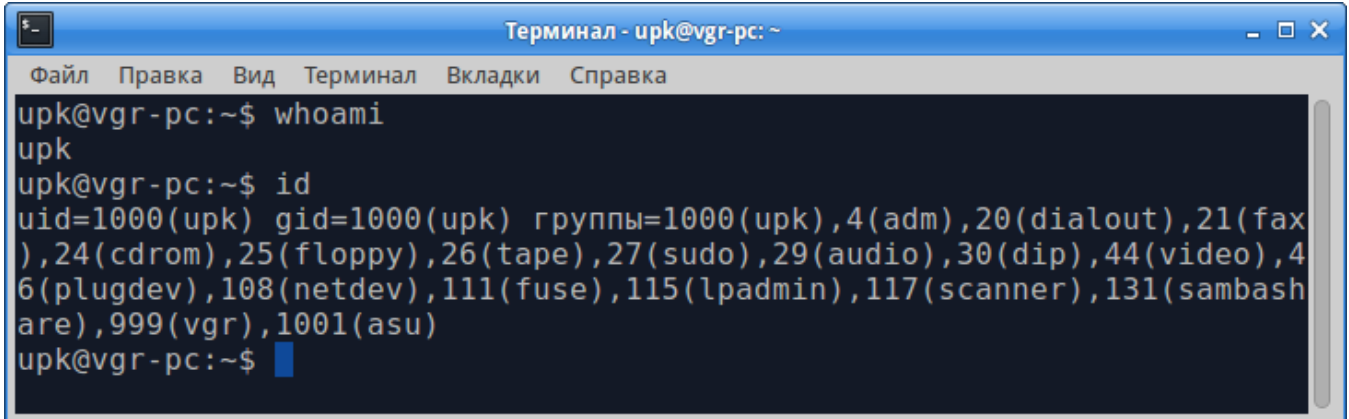
- *дочерний процесс*, создаваемый системным вызовом **fork(...)**, получает эффективные **ID** от своего родителя;
- *процесс*, модифицируемый одним из системных вызовов **exec(...)**, устанавливает эффективные **ID** в зависимости от значений битов **SUID** и **SGID**, присутствующих в поле *i\_mode* индексного дескриптора 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 устанавливаются значения действительных (реальных) **ID** пользователя, запустившего процесс.

**Сохранённые ID пользователя и ID группы** — это числовые (двухбайтовые) значения *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 эффективных ID выполняется с помощью системных вызовов `setuid(...)` и `setgid(...)`.

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



```

Терминал - upk@vgr-pc: ~
Файл Правка Вид Терминал Вкладки Справка
upk@vgr-pc:~$ whoami
upk
upk@vgr-pc:~$ id
uid=1000(upk) gid=1000(upk) группы=1000(upk),4(adm),20(dialout),21(fax),
24(cdrom),25(floppy),26(tape),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),111(fuse),115(lpadmin),117(scanner),131(sambashare),999(vgr),1001(asu)
upk@vgr-pc:~$
  
```

Рисунок 1.1 — Вывод эффективных ID пользователя upk

## 1.3 Login и система доступа Linux-PAM

**Любому пользователю**, для нормальной работы с ОС, необходимо пройти *процедуру 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 файлы: `/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 «*Менеджер сеанса*», который контролирует и обеспечивает процедуры входа пользователя в систему:

- *в случае положительного завершения* процедуры входа в систему, дополнительно контролируется содержимое директории `/home/$USER` и, случае необходимости, в нее добавляются нужные файлы или модифицируются старые, а также *устанавливаются эффективные идентификаторы* для процессов сеанса пользователя;
- *в случае негативного завершения* процедуры входа в систему или *негативного контроля* содержимого директории `/home/$USER`, выполняются дополнительные процедуры, которые заканчиваются перезапуском «*Менеджера сеансов*».

### Замечание

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

**Процедура входа в ОС** может быть: *текстовой*, когда на консоль терминала выводится приглашение *login:*, или *графической*, когда выводится некоторое стилизованное 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 «*Менеджера сеансов*».

*После ввода необходимой информации* начинается с *процедура login*, которая подразделяется на:

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

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

Поскольку методы авторизации могут быть различными, а пользователю даже приходится обращаться к программам, требующим смены пользователя, то *смена парадигмы обеспечения безопасности работы ОС*, приводит к перезаписи большого 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 поставила 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 приложениями и различными протоколами и способами идентификации и авторизации.

**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 разделяет прикладную часть процессов от части, обеспечивающей защиту их функционирования, одновременно централизуя ПО защиты, обеспечивая мобильность его модификации и ускоряя внедрение новых технологий.

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

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

*Практическая реализация этой концепции* предполагает централизованное использование файлов конфигурации.

*Сейчас*, файлы конфигурации Linux-PAM находятся в директории */etc/pam.d*, в которой находятся файлы, как правило совпадающие с именами файлов приложений. Например, для программы *sudo* имеется 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 четырёх типов: <i>auth, account, session, passwd</i> .		
управляющий_флаг	Флаг, контролирующий поведение PAM в случае успешного или безуспешного результата работы модуля: <ul style="list-style-type: none"> <li>• <b>required</b> - успешное завершение работы модуля необходимо для успеха всего запроса. Об ошибочном завершении не будет сообщено до окончания работы всех модулей.</li> <li>• <b>requisite</b> — успешное завершение работы модуля необходимо для успеха всего запроса. Ошибочное завершение приводит к немедленному возврату управления приложению.</li> <li>• <b>sufficient</b> — в случае успешного завершения, управление немедленно передаётся приложению. Ошибочное завершение модуля не учитывается.</li> <li>• <b>optional</b> — результат работы этого модуля не учитывается.</li> </ul>		

имя_модуля	Имя файла, которое должно быть указано с полным путём 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)
$
```

## 1.4 Команды управления пользователями

*Теоретическая концепция* пользователя поддерживается соответствующей информационной и управляющей инфраструктурой ОС.

*Хотя работа с пользователями* предполагает всего три действия: *создание*, *удаление* 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:

```
userdel имя_пользователя;
```

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

- *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 ориентированы на изменение соответствующих параметров.

Особое место в управлении пользователями занимает утилита *passwd*, которая управляет 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 её с ключём —*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
$
```

## 2 Лабораторная работа №5

*Цель лабораторной работы №5* — практическое закрепление учебного материала по теме «Управление пользователями ОС».

*Метод достижения указанной цели* — закрепление учебного материала, изложенного в первом разделе пособия посредством утилит ОС, а также выполнение заданий, приведённых в данном разделе.

*Чтобы успешно выполнить данную работу*, студенту следует:

- *запустить с flashUSB* ОС УПК АСУ, подключить личный архив и переключиться в сеанс пользователя *upk*;
- *запустить на чтение* данное пособие и на редактирование личный отчёт;
- *открыть одно или несколько окон терминалов*, причём хотя бы в одном окне терминала открыть Midnight Commander, для удобства работы с файловой системой ОС;
- *приступить к выполнению работы*, последовательно пользуясь рекомендациями представленных ниже подразделов.

### Замечание

Многие команды ОС студенту ещё не известны, поэтому следует:

- для вывода на консоль руководства по интересующей команде, использовать: *man имя\_команды*;
- для выяснения существования команды, ее доступности и местоположения, использовать: *command -v имя\_команды*;
- для уточнения правил запуска конкретной команды, можно попробовать один из вариантов: *команда --help* или *команда -h* или *команда -?*.

*В процессе выполнения лабораторной работы студент заполняет личный отчёт по каждому изученному вопросу!*

### 2.1 Инфраструктура управления пользователями

Прочитайте и усвойте учебный материал подраздела 1.1.

Исследуйте содержимое директорий: */etc/passwd*, */etc/shadow*, */etc/group*, */etc/gshadow*.

Усвойте структуру и назначение этих файлов.

### 2.2 Реальные и эффективные права пользователя

Прочитайте и усвойте учебный материал подраздела 1.2.

С помощью руководства *man* изучите утилиты *whoami*, *id*, *chown* и *chmod*.

С помощью пособия [8, подраздел 1.5, таблицы 1.9 и 1.10] изучите структуру поля дескриптора файлов *i\_mode*.



Усвойте назначение битов **SUID** и **SGID**.

В директории `~/src` создайте текстовый файл **test** и включите в него команды **id** и **whoami**.

Сделайте файл `~/src/test` исполняемым и, запуская его, исследуйте эффективные идентификаторы запускаемого процесса.

Находясь в директории `~/src`, установите значения битов **SUID** и **SGID** командой:

```
sudo chmod 3777 ./test
```

Запустите команды:

```
./test
sudo ./test
```

Сравните результаты.

## 2.3 Инфраструктура PAM

Прочитайте и усвойте учебный материал подраздела 1.3.

Исследуйте содержимое директории `/etc/skel` и сравните с содержимым рабочей директории пользователя **upk**.

Изучите содержимое директории `/lib/security`.

Изучите содержимое директории `/etc/pam.d` и файла `/etc/pam.conf`.

Изучите утилиту **ldd** и исследуйте с помощью нее ряд утилит, которые вы считаете, участвуют в контроле прав доступа пользователей.

## 2.4 Команды управления пользователями

Прочитайте и усвойте учебный материал подраздела 1.4.

Изучите утилиты **useradd**, **userdel** и **usermod**.

### Замечание

В данном варианте дистрибутива не установлено графическое приложение для работы с пользователями ОС, поэтому задание, изложенное ниже, следует выполнить пользуясь только утилитами **useradd**, **userdel** и **usermod**.

Из главного меню рабочего стола откройте окно «**Все настройки**», выберите и запустите ПО, озаглавленное «**Пользователи и группы**», как показано на рисунке 2.1.

Добавьте нового пользователя, например с именем **mmm**.

Исследуйте содержимое директорий: `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`.

Запустите *Midnight Commander* в окне терминала и перейдите в директорию `/home/mmm`.

Сравните содержимое директории `/home/mmm` с содержимым директории `/etc/skel`.

Закройте все окна пользователя *upk* и выйдите из его сеанса.

Войдите в сеанс пользователя *mmm*, запустите *Midnight Commander* в окне терминала и исследуйте содержимое директории `/home/upk`.

Выйдите из сеанса пользователя *mmm* и зайдите в сеанс пользователя *upk*.

Запустите главное окно работы с пользователями и удалите пользователя *mmm*.

Исследуйте изменения структуры файлов и директорий.

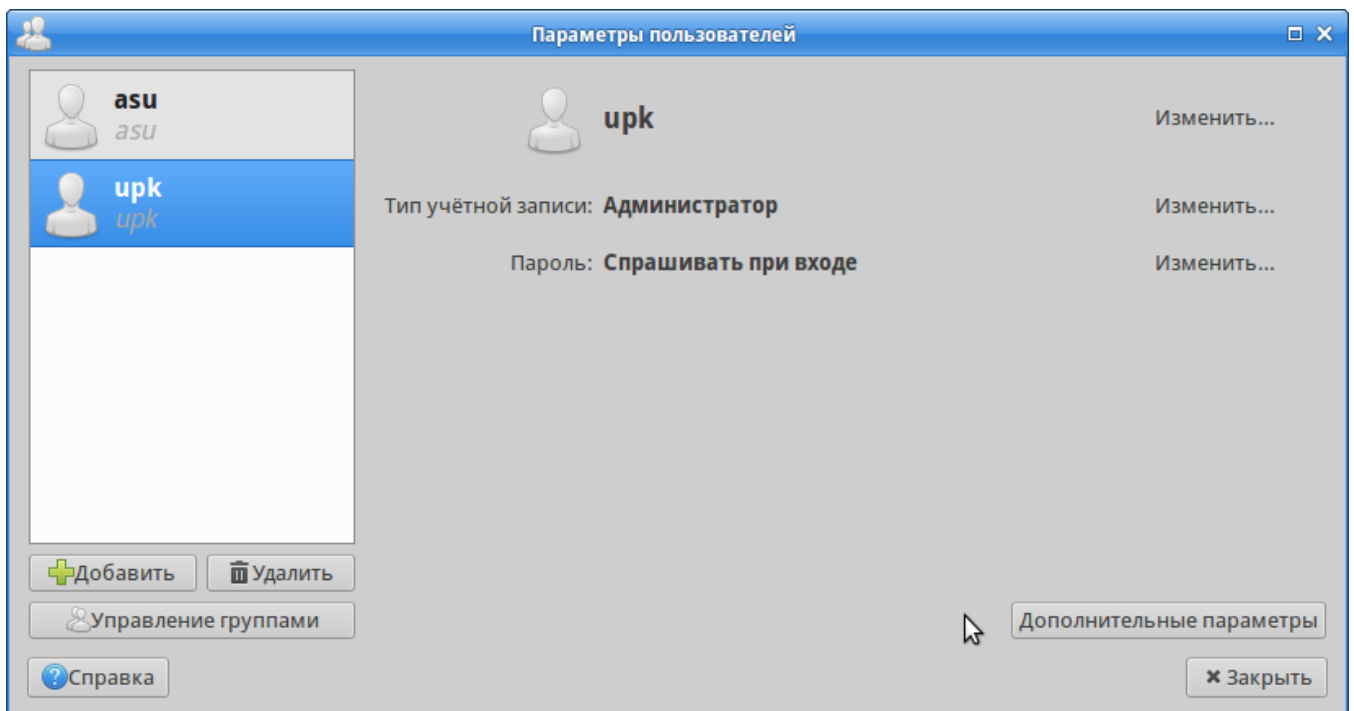


Рисунок 2.1 — Главное окно работы с пользователями ОС

Завершив выполнение всех заданий и оформление отчёта:

- провести архивирование и сохранение рабочей области *upk*;
- выключить компьютер и завершить выполнение лабораторной работы №5.

## **Список использованных источников**

- 1 Резник В.Г. Операционные системы. Самостоятельная и индивидуальная работа студента по направлению подготовки бакалавра 09.03.03. Учебно-методическое пособие. – Томск, ТУСУР, 2016. – 13 с.
- 2 Гордеев А.В. Операционные системы: учебное пособие для вузов. — СПб.: Питер, 2004. — 415с.
- 3 Таненбаум Э. Современные операционные системы. 4-е изд. - СПб.: Питер, 2015. - 1120с.
- 4 Резник В.Г. Учебный программный комплекс кафедры АСУ на базе ОС ArchLinux. Учебно-методическое пособие. – Томск, ТУСУР, 2017. – 38 с.
- 5 Резник В.Г. Операционные системы. Тема 1. Назначение и функции ОС. Учебно-методическое пособие. – Томск, ТУСУР, 2017. – 33 с.
- 6 Резник В.Г. Операционные системы. Тема 2. BIOS, UEFI и загрузка ОС. Учебно-методическое пособие. – Томск, ТУСУР, 2017. – 30 с.
- 7 Резник В.Г. Операционные системы. Тема 3. Языки управления ОС. Учебно-методическое пособие. – Томск, ТУСУР, 2017. – 38 с.
- 8 Резник В.Г. Операционные системы. Тема 4. Управление файловыми системами ОС. Учебно-методическое пособие. – Томск, ТУСУР, 2017. – 48 с.