

# Операционные системы семейства UNIX

Выполнил студент Владимир Горычев  
2014 год

# Содержание

Краткие сведения о развитии ОС UNIX

Версии UNIX

Бесплатные версии ОС семейства UNIX

Общие черты UNIX-систем

Управление компьютером под управлением UNIX

Процессы в ОС UNIX

Атрибуты процесса

Средства системного администрирования

Учетные записи пользователей

Ядро операционной системы

Специальные символы командной оболочки

Интерпретация командной строки

Переменные оболочки

Структура файловой системы

Структура файловой системы UNIX

Типы файлов ОС UNIX

# Краткие сведения о развитии ОС UNIX

- *ОС UNIX появилась в конце 60-х годов, как операционная система для мини-ЭВМ PDP-7. Активное участие в разработке приняли Кеннет Томсон и Деннис Ритчи.*
  - *Особенностями ОС UNIX стали: многопользовательский режим, новая архитектура файловой системы и др.*
- *В 1973 году большая часть ядра ОС была переписана на новом языке C.*
- *С 1974 года ОС UNIX распространяется в университетах США в исходных кодах.*

# Версии UNIX

- С самого начала распространения UNIX в американских университетах начали появляться различные версии ОС.
- Для упорядочивания фирма AT&T в 1982 объединила несколько версий в одну назвала вариант ОС – System III. В 1983 году была выпущена коммерческая версия – System V. В 1993 году AT&T продала свои права на UNIX фирме Novell, которая далее консорциуму X/Open и Santa Cruz Operation (SCO).
- Другая линия ОС UNIX – BSD разрабатывается в Калифорнийском университете (Беркли). Существуют бесплатные версии FreeBSD, OpenBSD.



# Бесплатные версии ОС семейства UNIX

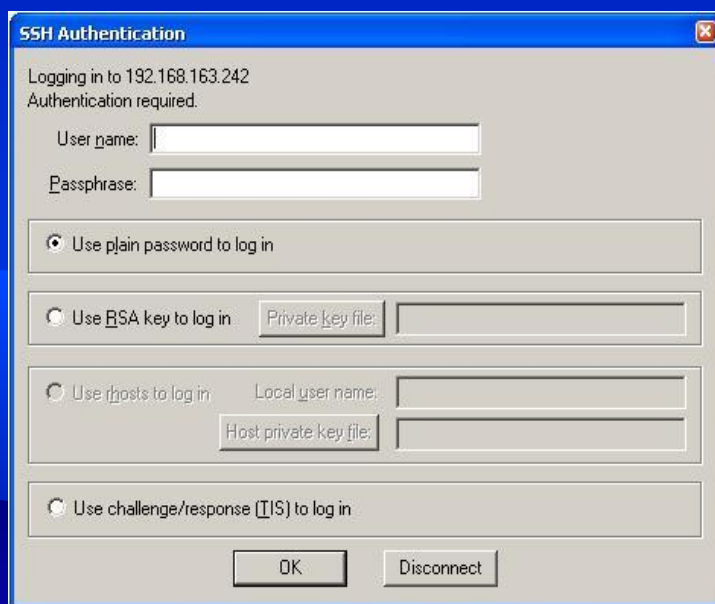
- Существует большое количество бесплатных версий UNIX.
  - FreeBSD, NetBSD, OpenBSD – варианты, разрабатываемые на основе ОС BSD.
- Наиболее популярное семейство бесплатных UNIX-систем – это системы семейства Linux. Первый вариант Linux был разработан Линусом Торвальдсом в 1991 г. В настоящее время существует несколько вариантов Linux: Red Hat, Mandrake, Slackware, SuSE, Debian.

# Общие черты UNIX-систем

- Различные варианты UNIX обладают рядом общих черт:
  - Мультипрограммная обработка в режиме разделения времени, основанная на вытесняющей многозадачности;
  - Поддержка многопользовательского режима;
    - Использование механизмов виртуальной памяти и свопинга;
    - Иерархическая файловая система;
  - Унификация операций ввода/вывода на основе расширенного использования понятия файл;
    - Переносимость системы;
  - Наличие сетевых средств взаимодействия.

# Управление компьютером под управлением UNIX

- При работе с UNIX-системой в режиме сервера, как правило, используется режим удаленного доступа с помощью некоторой терминальной программы.
  - Сеанс работы начинается с ввода регистрационного имени и пароля доступа.



# Процессы в ОС UNIX

- Процесс в ОС создается в процессе запуска приложения со стороны пользователя или самой ОС. Для каждого процесса ОС характерны совокупность набора команд процессора и ассоциированных ресурсов – адресное пространство, стеки, используемые файлы и устройства ввода-вывода и т.п.
- Многозадачность ОС означает, что одновременно исполняется множество процессов и задача операционной системы корректно распределить множество имеющихся ресурсов.
  - Различают:
    - **независимые процессы** – используют ресурсы, но не обмениваются информацией;
    - **взаимодействующие процессы** – обмениваются информацией, либо их выполнение синхронизировано.
    - Процессы взаимодействуют с помощью специальных механизмов:
      - сигналы;
      - программные каналы;
      - разделяемая память;
        - семафоры;
        - сообщения;
      - общие файлы.



# Атрибуты процесса

- Каждый процесс характеризуется набором атрибутов. К их числу относятся:
  - PID – идентификатор процесса
  - PPID – идентификатор родительского процесса
  - UID, GID – идентификаторы пользователя и группы
  - TT – управляющий терминал (процессы не связанные с управляющими терминалами называются демонами)
  - SID – идентификатор сессии, устанавливается равным PID лидера сессии;
  - NICE – приоритет процесса (относительный приоритет)
    - TIME – процессорное время.

# Атрибуты процесса

## Атрибуты процесса

- Таблица, содержащая список процессов имеет примерно следующий вид:

USER	PID	%CPU	%MEM	VSZ	RSS	TT	STAT	STARTED	TIME	COMMAND
<u>dima</u>	1731	0.0	1.6	1080	932	p0	R+	3:15PM	0:00.00	<u>-bash (bash)</u>
<u>root</u>	1	0.0	0.4	552	212	??	ILs	Tue12PM	0:00.04	<u>/sbin/init--</u>
<u>root</u>	2	0.0	0.0	0	0	??	DL	Tue12PM	0:00.31	<u>(pagedaemon)</u>
<u>root</u>	3	0.0	0.0	0	0	??	DL	Tue12PM	0:00.00	<u>(vmdaemon)</u>
<u>root</u>	4	0.0	0.0	0	0	??	DL	Tue12PM	0:01.24	<u>(bufdaemon)</u>
<u>root</u>	5	0.0	0.0	0	0	??	DL	Tue12PM	0:01.81	<u>(vnlru)</u>
<u>root</u>	6	0.0	0.0	0	0	??	DL	Tue12PM	1:35.73	<u>(syncer)</u>
<u>root</u>	60	0.0	0.4	448	248	??	Ss	Tue12PM	0:21.35	<u>/sbin/natd -u -m -</u>
<u>root</u>	76	0.0	0.9	944	544	??	Is	Tue12PM	0:01.17	<u>/usr/sbin/syslogd</u>
<u>root</u>	87	0.0	1.1	1076	620	??	Is	Tue12PM	0:00.02	<u>/usr/sbin/inetd -w</u>
<u>root</u>	89	0.0	1.0	996	592	??	Is	Tue12PM	0:01.39	<u>/usr/sbin/cron</u>
<u>root</u>	91	0.0	2.4	2740	1404	??	Is	Tue12PM	0:04.09	<u>/usr/sbin/sshd</u>
<u>root</u>	94	0.0	2.8	2788	1664	??	Ss	Tue12PM	0:14.07	<u>sendmail: acceptin</u>
<u>smmsp</u>	97	0.0	2.6	2660	1564	??	Is	Tue12PM	0:00.27	<u>sendmail: Queue ru</u>
<u>drweb</u>	217	0.0	3.6	2652	2132	??	Is	Tue12PM	0:00.00	<u>/usr/local/drweb/d</u>
<u>drweb</u>	222	0.0	1.1	1380	640	??	Ss	Tue12PM	0:06.06	<u>/usr/local/sbin/dr</u>
<u>root</u>	227	0.0	0.9	948	532	v1	Is+	Tue12PM	0:00.02	<u>/usr/libexec/getty</u>

# Средства системного администрирования

- Для управления операционной системой в UNIX часто используются конфигурационные файлы. Такие файлы определяют параметры запуска многих системных процессов.
- Для размещения конфигурационных файлов, как правило, используется каталог */etc*.
  - *adduser.conf* – определяет параметры пользователя
    - *crontab* – задает таблицу расписаний
    - *fstab* – определяет таблицу разделов
  - *ftpusers* – определяет параметры пользователей ftp
  - *hosts* – определяет список соответствий имен и ip-адресов
  - *hosts.allow* – определяет список разрешенных хостов
  - *rc.conf* – определяет конфигурацию сетевых подключений
    - и др.
- Многие файлы представляют собой сценарии, обрабатываемые оболочками.
- FreeBSD использует стартовый сценарий */etc/rc* и команды, включенные в него */etc/rc.conf* при запуске системы.

# Учетные записи пользователей

- Для упорядочивания работы с пользователями, хранения информации о их персональных настройках используются **учетные записи пользователей**.
- **Группа пользователей** – именованное объединение нескольких учетных записей. Группа может быть использована для разграничения доступа к данным.
- Имена групп и пользователей в текстовом виде используются для удобства самих пользователей. Система вместо имени использует идентификаторы:
  - UID – идентификатор пользователя;
  - GID – идентификатор группы.

# Ядро операционной системы

- Ядро ОС – низкоуровневая программа компьютера. Для большинства устройств, ядро – единственная программа, имеющая доступ. Доступ к устройствам осуществляется с помощью драйверов.
- Ядро реализует файловую систему, управляет памятью, контролирует выполнение программ, управляет доступом к сети.
- Ядро создается при инсталляции системы и хранится в специальном файле `/kernel`
  - При загрузке системы можно изменить параметры ядра:
    - `boot [-опции] [ядро]`
  - При необходимости можно загрузить отдельные модули ядра:
    - `kldload [модуль]`
    - выгрузка
    - `kldunload [модуль]`
- Список модулей хранится в специальном каталоге, например, `/modules`

# Ядро операционной системы

- При изменении параметров системы возможно потребуется изменение ядра ОС. Процесс создания нового ядра называется **компиляцией**.
- В процессе подготовки к компиляции подготавливается исходный код (дистрибутивы обычно хранятся в каталоге /usr/src) и создается конфигурационный файл (GENERIC).
  - Сам процесс компиляции включает три команды:
    - `cd /usr/src`
    - `make buildkernel KERNCONF=конф_файл`
    - `make installkernel KERNCONF=конф_файл`
- При компиляции старое ядро переименовывается в файл `kernel.old`, а файл `kernel` содержит новое ядро.
  - Загрузка нового ядра произойдет при следующей перезагрузке.

# Специальные символы командной оболочки

- При использовании командной оболочки, можно использовать некоторые специальные символы:
  - ; - используется для задания **списка команд**, обрабатываемых последовательно
  - | - используется для задания конвейера, используется для перенаправления стандартного вывода первой команды, на стандартный ввод второй
  - > - используется для перенаправления стандартного вывода программы в указанный файл
  - < - используется для перенаправления стандартного ввода

# Интерпретация командной строки

- При вводе команд используются следующие соглашения:
- Первое слово в командной строке является **именем команды**;
  - Остальные слова – **аргументы**.
    - Среди аргументов выделяются ключи (опции) – predetermined для каждой команды слова (символы), начинающиеся с одной (краткий формат) или пары дефисов (длинный формат). Например:
      - -bash-2.05b\$ tar -c -f arch.tar \*.c
      - -bash-2.05b\$ tar - -create - -file=arch.tar \*.c
    - При задании опций они могут объединяться. Например следующие команды равноправны:
      - -bash-2.05b\$ ls -a -l
      - -bash-2.05b\$ ls -l -a
      - -bash-2.05b\$ ls -al
  - Другие аргументы указывают на объекты, над которыми выполняются операции.



# Переменные оболочки

- При работе в системе существует способ передачи параметров в программы, кроме использования ключей командной оболочки, – использование **переменных окружения**. Для задания переменной окружения используется команда **set**. Формат команды:
  - `-bash-2.05b$ set имя_переменной=значение`
- Удаление переменной окружения выполняется командой **unset**.
  - Для обращения к значению переменной используется обозначение `$имя_переменной`, например команда:
    - `-bash-2.05b$ echo $PATH`
    - - выведет значение переменной `PATH`.

# Структура файловой системы

- **Загрузочный блок** – один или несколько секторов в начале раздела. Хранит название раздела и программный код, при старте запускающий ядро ОС в память компьютера.
- **Суперблок** – содержит общую информацию о файловой системе как совокупности файлов на данном разделе жесткого диска (размер раздела UNIX, число свободных и занятых блоков и индексных дескрипторов, флаг целостности файловой системы).
- **Таблица индексных дескрипторов** хранит дескрипторы файлов. Дескриптор содержит сведения о файле, размещении файла на диске, права доступа к файлу, UID владельца файла, GID группы файла, временные метки файла.
- **Файлы и каталоги.** Каталоги образуют древовидную структуру.

# Структура файловой системы UNIX

- Файловая система UNIX имеет иерархическую структуру, расширяемую до сетевой с помощью ссылок.
- В системе имеется единственный **корневой каталог** обозначаемый специальным символом /. Все прочие каталоги размещаются в нем или его подкаталогах.
  - Некоторые каталоги системы:
    - /bin – содержит часто употребляемые команды;
    - /lib – содержит файлы библиотек Си;
  - /home – содержит домашние каталоги пользователей;
  - /usr – содержит подкаталоги с файлами совместного использования;
    - /var – содержит временные файлы;
    - /sbin – содержит системные утилиты;
    - /root – домашний каталог супервизора;
  - /etc – содержит конфигурационные файлы системы;
  - /dev – содержит специальные файлы устройств.

# Типы файлов ОС UNIX

- В ОС используются следующие типы файлов:
  - **обычный файл** – используются для хранения информации;
  - **каталог** – содержит имена файлов, относящихся к данному каталогу;
  - **специальный файл устройства** – обеспечивает доступ к некоторому устройству;
  - **именованный канал** – используется для обмена данными между процессами;
    - **ссылка** – для создания связи между данными, хранящимися в разных файлах;
    - **сокеты** – для организации обмена данных между процессами, существует только пока на него ссылается хотя бы один процесс.