



Особенности операционных систем семейства 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.
- Особенности 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.
- К семейству OSF/1 – Open Software Foundation – относятся ОС консорциума IBM, DEC и Hewlett Packard. К числу ОС данного семейства относятся – HP-UX, AIX, Digital UNIX.

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.**

Достоинства UNIX-систем

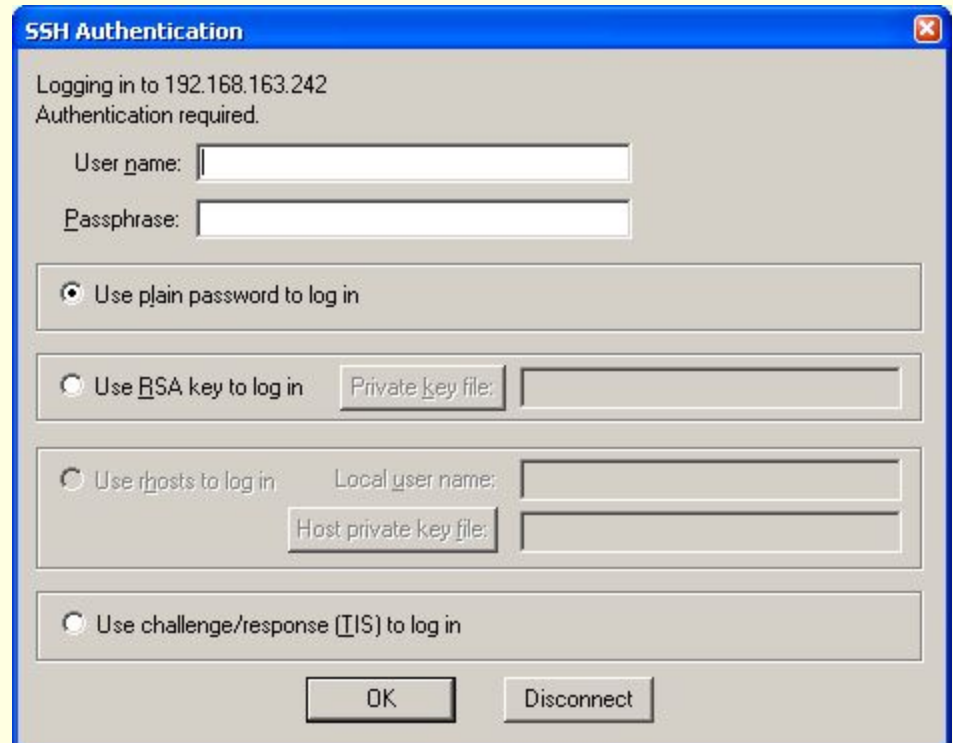
- К числу достоинств ОС семейства UNIX относят:
 - **Переносимость;**
 - **Эффективная реализация многозадачности;**
 - **Открытость;**
 - **Наличие и строгое соблюдение стандартов;**
 - **Единая файловая система;**
 - **Мощный командный язык;**
 - **Наличие значительного числа программных продуктов;**
 - **Реализация стека протокола TCP/IP;**
 - **Возможность работы в роли сервера или рабочей станции.**

Серверы на основе UNIX

- **Сервер** – компьютер, обрабатывающий запросы других компьютеров в сети и предоставляющий собственные ресурсы для хранения, обработки и передачи данных. Сервер под управлением UNIX может выполнять следующие роли:
 - Файловый сервер;
 - Web-сервер;
 - Почтовый сервер;
 - Сервер дистанционной регистрации (аутентификации);
 - Вспомогательные серверы Web-служб (DNS, DHCP);
 - Сервер доступа к сетям Интернет.

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.



Управление компьютером под управлением UNIX

- Часто для решения задач управления сервером ограничиваются командным режимом работы. В этом случае, для управления используется ввод специальных команд в **командную строку** в специальном формате. Командная строка имеет специальное приглашение, например:
 - `-bash-2.05b$`
- Общий вид команды:
 - `-bash-2.05b$ команда [опции] [параметры]`
- Например, вызов справки по ОС имеет вид:
 - `-bash-2.05b$ man [ключи] [тема]`
 - Для вызова справки по использованию команды `man` введите
 - `-bash-2.05b$ man man`

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) – предопределенные 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.

Примеры переменных окружения, используемых в командной оболочке bash

- Ряд переменных окружения определены в командных оболочках. Примеры таких переменных в bash:
 - 0 – имя оболочки или исполняющегося командного файла
 - # - число аргументов
 - номер – аргумент командной строки с соответствующим номером
 - ? – код завершения предыдущем команды
 - PS1 – формат приглашения командной строки
 - PATH – содержит перечень каталогов, в которых выполняется поиск исполняемых файлов
 - HOME – содержит полное имя домашнего каталога
 - PWD – содержит полное имя текущего каталога
 - USER – содержит имя пользователя

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

Файловая система ОС UNIX

- **Файл** – именованная область на внешнем носителе данных, используемая для хранения некоторой информации. В UNIX файл имеет более общее толкование. Файл – источник данных, которые могут быть считаны, или объект, куда могут быть записаны данные. В качестве файлов могут выступать, например, клавиатура или принтер, подключенный к параллельному порту.
- Для упорядочивания операций с файлами используется понятие каталога – структуры, объединяющей группу файлов и других каталогов.
- Файловая система – иерархическая структура, описывающая схему подчинения файлов и каталогов, размещаемых на носителе.

Особенности файловой системы FreeBSD

- FreeBSD использует используется файловая система FFS (Fast File System), по-другому – UFS. Свойства файловой системы:
 - управление владельцем файла;
 - управление правами доступа;
 - использование специальных флагов (например, указание типа файла – каталог, ссылка и т.п.);
 - управление характеристиками файлов (имена файлов, временные отметки и т.п.).

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 диске, права доступа к файлу, UID владельца файла, GID группы файла, временные метки файла.
- **Файлы и каталоги.** Каталоги образуют древовидную структуру.

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;
 - **сокет** – для организации обмена данных между процессами, существует только пока на него ссылается хотя бы один процесс.

Монтирование файловой системы

- Доступ к разделу на носителе информации обеспечивается **монтированием** раздела в общую файловую систему.
- Монтирование обозначается определением файла устройства и точкой монтирования.
- При старте системы программа **mount** запускается стартовым скриптом и автоматически монтирует системы указанные в файле `/etc/fstab`.
- Команда монтирования раздела:
 - `mount файл_устройства точка_монтирования`

Атрибуты файла

- Для управления доступом к файлу используются специальные атрибуты, определяющие права доступа – биты доступа. Класс доступа задается числовым идентификатором, определяющимся следующим образом:
 - Для каждой категории задается трехзначное двоичное число:
 - старший разряд определяет право на чтение;
 - второй разряд определяет право на запись;
 - младший разряд – на выполнение данного файла.
 - Категории определяются следующим образом:
 - первая категория – владелец файла
 - вторая категория – группа владельца
 - третья категория – остальные пользователи

Владелец			Группа			Остальные			Значение
R	W	X	R	W	X	R	W	X	
1	1	1	1	0	1	0	0	0	750

Команды манипулирования файлами

- Базовые операции с файлами:
 - `cd` – изменение текущего каталога;
 - `ls` – вывод списка файлов;
 - `cp` – копирование файлов;
 - `mv` – перемещение файлов;
 - `rm` – удаление файлов;
 - `mkdir` – создание каталога;
 - `rmdir` – удаление каталога;
 - `ln` – создание ссылки на файл;
 - `chmod` – команда изменения атрибутов;
 - `chown` – команда изменения собственника.

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.

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

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

Команды управления процессами

- Существует ряд команд, позволяющих просматривать и управлять процессами в системе:
 - ps – выводит информацию о выполняющихся процессах;
 - top – выводит и динамически обновляет список наиболее активных процессов;
 - nice – явно устанавливает приоритет процесса;
 - renice – корректирует приоритет процесса;
 - kill – завершение работы заданного процесса;
 - killall – завершение работы всех процессов, соответствующих заданному имени.

Выполнение задач в определенное время

- Существуют ситуации, когда выполнение заданий должно быть произведено в назначенное время в отсутствие системного администратора.
- Для выполнения команды в определенное время используются команды `at` или `batch`:
 - `at [-V] [-q очередь] [-f файл] [mldbv] время`
 - `-V` отображает номер версии программы
 - `-q очередь` поддерживается очередь заданий
 - `-f файл` команда задается в файле
 - `-m` задает отправку почтового сообщения пользователю
 - `-l` задает вывод списка заданий в очереди
 - `-d` задает удаление задания из очереди
 - `-v` выводит список выполненных заданий в очереди
 - `время` задает время для исполнения (возможно использование `+` для относительного времени).

Выполнение задач в определенное время

- Для выполнения задания периодически используется демон `cron`:
 - управляющий файл имеет имя `/etc/crontab`
 - Структура файла представлена ниже:
 - *минута час день_месяца месяц день_недели пользователь команда*
 - * задает непрерывное выполнение
 - Следующая запись
 - `05,35 8-17 * * mon-fri mailnull fetchmail`
 - Задает выполнение команды `fetchmail` от имени пользователя `mailnull` дважды в час (в 5 и 35 минут), с 8:05 до 17:35 с понедельника по пятницу
 - Планирование рядовыми пользователями осуществляется с помощью команды **`crontab`**, синтаксисом:
 - **`crontab -u user [-l | -r | -e] [file]`**
 - **`-l`** – вывести текущее расписание
 - **`-r`** – удалить текущее расписание
 - **`-e`** – редактировать текущее расписание

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 используем идентификаторы:
 - UID – идентификатор пользователя;
 - GID – идентификатор группы.

Хранение информации об учетных записях

- Информация об учетных записях хранится в нескольких структурах данных:
 - `/etc/passwd` – файл, содержащий основную информацию обо всех учетных записях:
 - `logname:*:UID:GID:GECOS:HOME:SHELL`
 - `ivlev:*:1038:1038:ivlev:/home/ivlev:/usr/local/bin/bash`
 - `/etc/master.passwd` – файл хранящий информацию из `/etc/passwd` и, кроме того, хэшированные значения паролей и ряд других сведений;
 - `/etc/pwd.db` `/etc/spwd.db` – специальные файлы баз данных для хранения информации подобно `/etc/passwd`
 - начальный каталог – содержит полное имя каталога для хранения пользовательских данных

Управление учетными записями

- Для управления учетными записями пользователей используются команды:
 - `adduser` – добавляет пользователя в систему;
 - `rmuser` – удаляет пользователя из системы;
 - `passwd` – изменяет пароль пользователя;
 - `chpass` – изменяет параметры учетной записи, такие как пароль, срок действия, стандартный интерпретатор команд;
 - `pwd_mkdb` – команда внесения изменений в базы данных учетных записей, после внесения изменений в файл `/etc/master.passwd`

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 потребуются 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.