Федеральное агентство по образованию

Ростовская-на-Дону государственная академия

Сельскохозяйственного машиностроения

Кафедра: "Информационных и управляющих систем"

Контрольная работа

по операционным системам

на тему:

"**Операционная система Unix**"

Студентки заочного факультета ИС

Группы ЗИС 3–1

Руководитель:

кандидат технических наук доцент

г. Ростов-на-Дону

2008-09 уч.г.

**Содержание**

Введение

1. История создания, назначение

2. Функциональные характеристики

3. Особенности архитектуры ОС UNIX

4. Способы управления процессами и ресурсами

5. Условия эксплуатации

6. Достоинства и недостатки ОС UNIX

Заключение

Список использованной литературы

**Введение**

UNIX появилась в 1969 году. За 30 с лишним лет система стала довольно популярной и получила распространение на машинах с различной мощностью обработки, от микропроцессоров до больших ЭВМ, обеспечивая на них общие условия выполнения программ. Система делится на две части. Одну часть составляют программы и сервисные функции – это делает операционную среду UNIX такой популярной; данная часть ОС легко доступна пользователям, она включает такие программы, как командный процессор, обмен сообщениями, пакеты обработки текстов и системы обработки исходных текстов программ. Другая часть включает в себя собственно операционную систему, поддерживающую эти программы и функции.

UNIX – традиционно сетевая операционная система.

**1. История создания, назначение**

UNIX зародился в лаборатории Bell Labs фирмы AT&T более 30 лет назад. В то время Bell Labs занималась разработкой многопользовательской системы разделения времени MULTICS (Multiplexed Information and Computing Service) совместно с MIT и General Electric, но эта система потерпела неудачу. Bell Labs отказалась от участия в проекте MULTICS, что дало возможность одному из ее исследователей, Кену Томпсону, заняться поисковой работой в направлении улучшения операционной среды Bell Labs. Томпсон, а также сотрудник Bell Labs Денис Ритчи и некоторые другие разрабатывали новую файловую систему, многие черты которой вели свое происхождение от MULTICS. Для проверки новой файловой системы Томпсон написал ядро ОС и некоторые программы для компьютера GE-645, который работал под управлением мультипрограммной системы разделения времени GECOS. У Кена Томпсона была написанная им еще во времена работы над MULTICS игра "Space Travel" - "Космическое путешествие". Он запускал ее на компьютере GE-645, но она работала на нем не очень хорошо из-за невысокой эффективности разделения времени. Кроме этого, машинное время GE-645 стоило слишком дорого. В результате Томпсон и Ритчи решили перенести игру на машину PDP-7 фирмы DEC, имеющую 4096 18-битных слов, телетайп и хороший графический дисплей. Но у PDP-7 было неважное программное обеспечение, и, закончив перенос игры, Томпсон решил реализовать на PDP-7 ту файловую систему, над который он работал на GE-645. Из этой работы и возникла первая версия UNIX. Уже тогда она включала характерную для современной UNIX файловую систему, основанную на индексных дескрипторах inode, имела подсистему управления процессами и памятью, а также позволяла двум пользователям работать в режиме разделения времени. Система была написана на ассемблере. Имя UNIX (Uniplex Information and Computing Services) было дано ей одним сотрудником Bell Labs, Брайаном Керниганом,

Первыми пользователями UNIX'а стали сотрудники отдела патентов Bell Labs, которые нашли ее удобной средой для создания текстов. Большое влияние на судьбу UNIX оказала перепись ее на языке высокого уровня С, разработанного Денисом Ритчи специально для этих целей. Это произошло в 1973 году, UNIX насчитывал к этому времени уже 25 инсталляций, и в Bell Labs была создана специальная группа поддержки UNIX.

После описания системы Томпсоном и Ритчи в компьютерном журнале CACM в 1974 г. UNIX получил широкое распространение. ОС стала востребована в университетах, так как для них она поставлялась бесплатно вместе с исходными кодами на С. Широкое распространение эффективных C-компиляторов сделало UNIX уникальной для того времени ОС из-за возможности переноса на различные компьютеры. Университеты внесли значительный вклад в улучшение UNIX и дальнейшую его популяризацию. Еще одним шагом на пути к признанию UNIX, как стандартизованной среды стала разработка Денисом Ритчи библиотеки ввода-вывода stdio. Благодаря использованию этой библиотеки для компилятора С, программы для UNIX стали легко переносимыми.

ОС UNIX является интерактивной операционной системой, это традиционно сетевая операционная система.

**2. Функциональные характеристики**

К основным функциям ядра ОС UNIX принято относить следующие:

1. Инициализация системы - функция запуска и раскрутки. Ядро системы обеспечивает средство раскрутки (bootstrap), которое обеспечивает загрузку полного ядра в память компьютера и запускает ядро.
2. Управление процессами и нитями - функция создания, завершения и отслеживания существующих процессов и нитей ("процессов", выполняемых на общей виртуальной памяти). Поскольку ОС UNIX является мультипроцессной операционной системой, ядро обеспечивает разделение между запущенными процессами времени процессора (или процессоров в мультипроцессорных системах) и других ресурсов компьютера для создания внешнего ощущения того, что процессы реально выполняются в параллель.
3. Управление памятью - функция отображения практически неограниченной виртуальной памяти процессов в физическую оперативную память компьютера, которая имеет ограниченные размеры. Соответствующий компонент ядра обеспечивает разделяемое использование одних и тех же областей оперативной памяти несколькими процессами с использованием внешней памяти.
4. Управление файлами - функция, реализующая абстракцию файловой системы, - иерархии каталогов и файлов. Файловые системы ОС UNIX поддерживают несколько типов файлов. Некоторые файлы могут содержать данные в формате ASCII, другие будут соответствовать внешним устройствам. В файловой системе хранятся объектные файлы, выполняемые файлы и т.д. Файлы обычно хранятся на устройствах внешней памяти; доступ к ним обеспечивается средствами ядра. В мире UNIX существует несколько типов организации файловых систем. Современные варианты ОС UNIX одновременно поддерживают большинство типов файловых систем.
5. Коммуникационные средства - функция, обеспечивающая возможности обмена данными между процессами, выполняющимися внутри одного компьютера (IPC - Inter-Process Communications), между процессами, выполняющимися в разных узлах локальной или глобальной сети передачи данных, а также между процессами и драйверами внешних устройств.
6. Программный интерфейс - функция, обеспечивающая доступ к возможностям ядра со стороны пользовательских процессов на основе механизма системных вызовов, оформленных в виде библиотеки функций.

**3. Особенности архитектуры ОС UNIX**

Архитектура ОС UNIX – многоуровневая (рис.1). На нижнем уровне, непосредственно над оборудованием, работает ядро операционной системы. Функции ядра доступны через интерфейс системных вызовов, образующих второй уровень. На следующем уровне работают командные интерпретаторы, команды и утилиты системного администрирования, коммуникационные драйверы и протоколы, - все то, что обычно относят к системному программному обеспечению. Наконец, внешний уровень образуют прикладные программы пользователя, сетевые и другие коммуникационные службы, СУБД и утилиты.

**4. Способы управления процессами и ресурсами**

Файлы и процессы, являются центральными понятиями модели операционной системы UNIX. Рисунок 1.2 представляет блок-схему ядра системы, отражающую состав модулей, из которых состоит ядро, и их взаимосвязи друг с другом. Слева изображена файловая подсистема, а справа подсистема управления процессами – две главные компоненты ядра.

точка пересечения

Уровень пользователя

Библиотеки

Уровень ядра

Обращения к операционной системе

взаимодействие процессов

подсистема управления файлами

подсистема

управления

процессами

планировщик

буфер сверх-оперативной памяти(кеш)

символ блок

- - - - - - - - - - - - - - - - - - - - -

драйверы устройств

распределение памяти

аппаратный контроль

Уровень ядра аппаратуры

технические средства (аппаратура)

Рисунок.1.2 Блок-схема ядра операционной системы

Обращения к операционной системе выглядят так же, как обычные вызовы функций в программах на языке Си, и библиотеки устанавливают соответствие между этими вызовами функций и элементарными системными операциями. При этом программы на ассемблере могут обращаться к операционной системе непосредственно, без использования библиотеки системных вызовов. Программы часто обращаются к другим библиотекам, таким как библиотека стандартных подпрограмм ввода-вывода, достигая тем самым более полного использования системных услуг. Для этого во время компиляции библиотеки связываются с программами и частично включаются в программу пользователя. Совокупность обращений к операционной системе разделена на те обращения, которые взаимодействуют с подсистемой управления файлами, и те, которые взаимодействуют с подсистемой управления процессами. Файловая подсистема управляет файлами, размещает записи файлов, управляет свободным пространством, доступом к файлам и поиском данных для пользователей. Процессы взаимодействуют с подсистемой управления файлами, используя при этом совокупность специальных обращений к операционной системе, таких как open (для того, чтобы открыть файл на чтение или запись),close, read, write, stat (запросить атрибуты файла), chown (изменить запись с информацией о владельце файла) и chmod (изменить права доступа к файлу).

Подсистема управления файлами обращается к данным, которые хранятся в файле, используя буферный механизм, управляющий потоком данных между ядром и устройствами внешней памяти. Буферный механизм, взаимодействуя с драйверами устройств ввода-вывода блоками, инициирует передачу данных к ядру и обратно. Драйверы устройств являются такими модулями в составе ядра, которые управляют работой периферийных устройств. Устройства ввода-вывода блоками относятся программы пользователя к типу запоминающих устройств с произвольной выборкой; их драйверы построены таким образом, что все остальные компоненты системы воспринимают эти устройства как запоминающие устройства с произвольной выборкой. Например, драйвер запоминающего устройства на магнитной ленте позволяет ядру системы воспринимать это устройство как запоминающее устройство с произвольной выборкой. Подсистема управления файлами также непосредственно взаимодействует с драйверами устройств "неструктурированного" ввода-вывода, без вмешательства буферного механизма. К устройствам неструктурированного ввода-вывода, иногда именуемым устройствами посимвольного ввода-вывода (текстовыми), относятся устройства, отличные от устройств ввода-вывода блоками.

Подсистема управления процессами отвечает за синхронизацию процессов, взаимодействие процессов, распределение памяти и планирование выполнения процессов. Подсистема управления файлами и подсистема управления процессами взаимодействуют между собой, когда файл загружается в память на выполнение: подсистема управления процессами читает в память исполняемые файлы перед тем, как их выполнить.

Примерами обращений к операционной системе, используемых при управлении процессами, могут служить fork (создание нового процесса), exec (наложение образа программы на выполняемый процесс), exit (завершение выполнения процесса), wait (синхронизация продолжения выполнения основного процесса с моментом выхода из порожденного процесса), brk (управление размером памяти, выделенной процессу) и signal (управление реакцией процесса на возникновение экстраординарных событий.

Модуль распределения памяти контролирует выделение памяти процессам. Если в какой-то момент система испытывает недостаток в физической памяти для запуска всех процессов, ядро пересылает процессы между основной и внешней памятью с тем, чтобы все процессы имели возможность выполняться. Существует два способа управления распределением памяти: выгрузка (подкачка) и замещение страниц. Программу подкачки иногда называют планировщиком, т.к. она "планирует" выделение памяти процессам и оказывает влияние на работу планировщика центрального процессора. «Планировщик» планирует очередность выполнения процессов до тех пор, пока они добровольно не освободят центральный процессор, дождавшись выделения какого-либо ресурса, или пока ядро системы не выгрузит их после того, как их время выполнения превысит заранее определенный квант времени. Планировщик выбирает на выполнение готовый к запуску процесс с наивысшим приоритетом; выполнение предыдущего процесса (приостановленного) будет продолжено тогда, когда его приоритет будет наивысшим среди приоритетов всех готовых к запуску процессов. Существует несколько форм взаимодействия процессов между собой, от асинхронного обмена сигналами о событиях до синхронного обмена сообщениями.

Наконец, аппаратный контроль отвечает за обработку прерываний и за связь с машиной. Такие устройства, как диски и терминалы, могут прерывать работу центрального процессора во время выполнения процесса. При этом ядро системы после обработки прерывания может возобновить выполнение прерванного процесса. Прерывания обрабатываются не самими процессами, а специальными функциями ядра системы, перечисленными в контексте выполняемого процесса.

**5. Условия эксплуатации**

UNIX - многопользовательская операционная система. Пользователи, занимающиеся общими задачами, могут объединяться в группы. Каждый пользователь обязательно принадлежит к одной или нескольким группам. Все команды выполняются от имени определенного пользователя, принадлежащего в момент выполнения к определенной группе.

В многопользовательских системах необходимо обеспечивать защиту объектов (файлов, процессов), принадлежащих одному пользователю, от всех остальных. ОС UNIX предлагает базовые средства защиты и совместного использования файлов на основе отслеживания пользователя и группы, владеющих файлом, трех уровней доступа (для пользователя-владельца, для пользователей группы-владельца, и для всех остальных пользователей) и трех базовых прав доступа к файлам (на чтение, на запись и на выполнение). Базовые средства защиты процессов основаны на отслеживании принадлежности процессов пользователям.

Для отслеживания владельцев процессов и файлов используются числовые идентификаторы. Идентификатор пользователя и группы - целое число (обычно) в диапазоне от 0 до 65535. Присвоение уникального идентификатора пользователя выполняется при заведении системным администратором нового регистрационного имени. Значения идентификатора пользователя и группы - не просто числа, которые идентифицируют пользователя, - они определяют владельцев файлов и процессов. Среди пользователей системы выделяется один пользователь - системный администратор или суперпользователь, обладающий всей полнотой прав на использование и конфигурирование системы. Это пользователь с идентификатором 0 и регистрационным именем root.

При представлении информации человеку удобнее использовать вместо соответствующих идентификаторов символьные имена - регистрационное имя пользователя и имя группы. Соответствие идентификаторов и символьных имен, а также другая информация о пользователях и группах в системе (учетные записи), как и большинство другой информации о конфигурации системы UNIX, по традиции, представлена в виде текстовых файлов. Эти файлы - /etc/passwd, /etc/group и /etc/shadow.

**6. Достоинства и недостатки**

Широкое распространение UNIX породило проблему несовместимости его многочисленных версий. Для пользователя весьма неприятен тот факт, что пакет, купленный для одной версии UNIX, отказывается работать на другой версии UNIX. Периодически делались и делаются попытки стандартизации UNIX, но они пока имеют ограниченный успех. Процесс сближения различных версий UNIX и их расхождения носит циклический характер. Перед лицом новой угрозы со стороны какой-либо другой операционной системы различные производители UNIX-версий сближают свои продукты, но затем конкурентная борьба вынуждает их делать оригинальные улучшения и версии снова расходятся. В этом процессе есть и положительная сторона - появление новых идей и средств, улучшающих как UNIX, так и многие другие операционные системы, перенявшие у него за долгие годы его существования много полезного. Наибольшее распространение получили две несовместимые линии версий UNIX: линия AT&T - UNIX System V, и линия университета Berkeley-BSD. Многие фирмы на основе этих версий разработали и поддерживают свои версии UNIX: SunOS и Solaris фирмы Sun Microsystems, UX фирмы Hewlett-Packard, XENIX фирмы Microsoft, AIX фирмы IBM, UnixWare фирмы Novell (проданный теперь компании SCO), и список этот можно еще долго продолжать.

Наибольшее влияние на унификацию версий UNIX оказали такие стандарты как SVID фирмы AT&T, POSIX, созданный под эгидой IEEE, и XPG4 консорциума X/Open. В этих стандартах сформулированы требования к интерфейсу между приложениями и ОС, что дает возможность приложениям успешно работать под управлением различных версий UNIX.

Одним из основных преимуществ семейства операционных систем типа UNIX и возникшего на их основе подхода к стандартизации интерфейсов операционных систем (важная часть общего подхода открытых систем) является то, что они обеспечивают единую операционную среду на компьютерах с разной архитектурой.

**Заключение**

Операционная система UNIX, являющаяся первой в истории мобильной ОС, обеспечивающей надежную среду разработки и использования мобильных прикладных систем, одновременно представляет собой практическую основу для построения открытых программно-аппаратных систем и комплексов. Именно широкое внедрение в практику ОС UNIX позволило перейти от лозунга Открытых Систем к практической разработке этой концепции. Большой вклад в развитие направления Открытых Систем внесла деятельность по стандартизации интерфейсов ОС UNIX.

Можно выделить несколько ветвей ОС UNIX, различающихся не только реализацией, но временами интерфейсами и семантикой (хотя, по мере развития процесса стандартизации, эти различия становятся все менее значительными). Сегодня популярен новый свободно распространяемый вариант ОС UNIX, называемый FreeBSD. Ведутся работы над более развитыми версиями BSDNet.

**Список использованной литературы**

1. История и общая характеристика семейства операционных систем UNIX http://www.osys.ru/
2. Эбен М., Таймэн Б. «FreeBSD. Энциклопедия пользователя» - К: ООО «ТИД «ДС», 2002.
3. Керниган Б.В., Пайк Р. «UNIX - универсальная среда программирования» - М.: Финансы и статистика, 1992.
4. С.Д. Кузнецов, «Операционная система UNIX» - статья в Интернете.