Лабораторная работа No 1 «Выполнение команд в среде ОС Linux

1. Цель работы

Целью работы является изучение архитектуры и принципов функционирования многопользовательской многозадачной операционной системы Linux, особенности ее использования в качестве рабочей станции.

2. Теоретическая часть

ОС Linux включает следующие основные компоненты.

Ядро. Выполняет функции управления памятью, процессорами. Осуществляет диспетчеризацию выполнения всех программ и обслуживание внешних устройств. Все действия, связанные с вводом/выводом и выполнением системных операций, выполняются с помощью системных вызовов. Системные вызовы реализуют программный интерфейс между программами и ядром. Имеется возможность динамического конфигурирования ядра.

Диспетчер процессов Init. Активизирует процессы, необходимые для нормальной работы системы и производит их начальную инициализацию. Обеспечивает завершение работы системы, организует сеансы работы пользователей, в том числе, для удаленных терминалов.

Интерпретатор команд Shell. Анализирует команды, вводимые с терминала либо из командного файла, и передает их для выполнения в ядро системы. Команды обычно имеют аргументы и параметры, которые обеспечивают модернизацию выполняемых действий. Shell является также языком программирования, на котором можно создавать командные файлы (shell-файлы). При входе в ОС пользователь получает копию интерпретатора Shell в качестве родительского процесса. Далее, после ввода команды пользователем создается порожденный процесс, называемый процессом- потомком. Т.е. после запуска ОС каждый новый процесс функционирует только как процесс - потомок уже существующего процесса. В ОС Linux имеется возможность динамического порождения и управления процессами.

Обязательным в системе является интерпретатор Bash, полностью соответствующий стандарту POSIX. В качестве Shell может быть использована оболочка mc с интерфейсом, подобным Norton Commander.

Сетевой графический интерфейс X-сервер (X-Windows).

Обеспечивает поддержку графических оболочек.  
Графические оболочки KDE, Gnome. Отличительными свойствами

KDE являются: минимальные требования к аппаратуре, высокая надежность, интернационализация. Базовые библиотеки KDE (qt, kde-libs) признаны одними из лучших продуктов по созданию графического интерфейса, обеспечивают простое написание программ с использованием передовых технологий. Gnome имеет развитые графические возможности, но более требователен к аппаратным средствам.

Сетевая поддержка NFS, SMB, TCP/IP. NFS - программный комплекс РС-NFS (Network File System) для выполнения сетевых функций. РС-NFS ориентирован для конкретной ОС персонального компьютера (PC) и включает драйверы для работы в сети и дополнительные утилиты. SMB - сетевая файловая система, совместимая с Windows NT. TCP/IP - протокол контроля передачи данных (Transfer Control Protocol/Internet Protocol). Сеть по протоколам TCP/IP является неотъемлемой частью ОС семейства UNIX. Поддерживаются любые сети, от локальных до Internet, с использованием только встроенных сетевых средств.

Инструментальные средства программирования. Основой средств программирования является компилятор СС или GCC для языков С и С++; модули поддержки других языков программирования (Obective C, Фортран, Паскаль, Modula-3, Ада, Java и др.); интегрированные среды и средства визуального проектирования: Kdevelop, Xwpe; средства адаптации привязки программ AUTOCONFIG, AUTOMAKE.

Выполнение простых команд

Формат команд в ОС LINUX следующий:

имя команды [аргументы] [параметры] [метасимволы]

Имя команды может содержать любое допустимое имя файла; аргументы - одна или несколько букв со знаком минус (-); параметры - передаваемые значения для обработки; метасимволы интерпретируются как специальные операции. В квадратных скобках указываются необязательные части команд.

Введите команду echo, которая выдает на экран свои параметры:

echo good morning

и нажмите клавишу Enter. На экране появится приветствие "good morning" – параметр команды echo. Командный интерпретатор shell вызвал команду echo, реализованную в виде программы на языке СИ, и передал ей параметры. После этого интерпретатор команд вывел знак-приглашение. Синтаксис команды echo:

echo [-n] [arg1] [arg2] [arg3]...

Команда помещает в стандартный вывод свои параметры, разделенные пробелами и завершаемые символом перевода строки. При наличии флага -n символ перевода строки исключается.

who [am i] - получение информации о работающих пользователях.  
В квадратных скобках указываются параметры команды, которые можно опустить. Ответ представляется в виде таблицы, которая содержит следующую информацию:

- идентификатор пользователя; - идентификатор терминала;  
- дата подключения;  
- время подключения.

date - вывод на экран текущей даты и текущего времени.

cal [[месяц]год] - календарь; если календарь не помещается на одном экране, то используется команда cal год | more и клавишей пробела

производится постраничный вывод информации.  
man <название команды> - вызов электронного справочника об

указанной команде. Выход из справочника - нажатие клавиши Q.  
Команда man man сообщает информацию о том, как пользоваться справочником.

tty - сообщение имени специального файла стандартного вывода, соответствующего терминалу пользователя.

cat <имя файла> - вывод содержимого файла на экран. Команда cat > text.1 создает новый файл с именем text.1, который можно заполнить символьными строками, вводя их с клавиатуры. Нажатие клавиши Enter создает новую строку. Завершение ввода - нажатие Ctrl - d. Команда cat text.1 > text.2 пересылает содержимое файла text.1 в файл text.2. Слияние файлов осуществляется командой cat text.1 text.2 > text.3.

ls [-alrstu] [имя] - вывод содержимого каталога на экран. Если параметр не указан, выдается содержимое текущего каталога.  
Аргументы команды:

-a - выводит список всех файлов и каталогов, в том числе и скрытых;

-l - выводит список файлов в расширенном формате, показывая тип каждого элемента, полномочия, владельца, размер и дату последней модификации;

- r - выводит список в порядке, обратном заданному;  
- s - выводит размеры каждого файла;  
- t - перечисляет файлы и каталоги в соответствии с датой их последней

модификации;  
- u - перечисляет файлы и каталоги в порядке, обратном их последней

модификации.  
rm <имя файла> - удаление файла (файлов). Команда rm text.1 text.2

text.3 удаляет файлы text.1, text.2, text.3. Другие варианты этой команды - rm text.[123] или rm text.[1-3].

wc [имя файла] - вывод числа строк, слов и символов в файле. clear - очистка экрана.

Группирование команд

Группы команд или сложные команды могут формироваться с помощью специальных символов (метасимволов):

& - процесс выполняется в фоновом режиме, не дожидаясь окончания предыдущих процессов;

? - шаблон, распространяется только на один символ;  
\* - шаблон, распространяется на все оставшиеся символы;  
| - программный канал - стандартный вывод одного процесса является

стандартным вводом другого;  
> - переадресация вывода в файл;  
< - переадресация ввода из файла;  
; - если в списке команд команды отделяются друг от друга точкой с

запятой, то они выполняются друг за другом;

&& - эта конструкция между командами означает, что последующая команда выполняется только при нормальном завершении предыдущей команды ( код возврата 0 );

|| - последующая команда выполняется только, если не выполнилась предыдущая команда ( код возврата 1 );

() - группирование команд в скобки;  
{ } - группирование команд с объединенным выводом;  
[] - указание диапазона или явное перечисление (без запятых); >> - добавление содержимого файла в конец другого файла.

Примеры.

who | wc - подсчет количества работающих пользователей командой wс (word count - счет слов);

cat text.1 > text.2 - содержимое файла text.1 пересылается в файл text.2;

mail student < file.txt - электронная почта передает файл file.txt всем пользователям, перечисленным в командной строке;

cat text.1,text.2 - просматриваются файлы text.1 и text.2;  
cat text.1 >> text.2 - добавление файла text.1 в конец файла text.2;  
cc primer.c & - трансляция Си - программы в фоновом режиме. Имя

выполняемой программы по умолчанию a.out.  
cc -o primer.o primer.c - трансляция Си-программы с образованием

файла выполняемой программы с именем primer.o;  
rm text.\* - удаление всех файлов с именем text;  
{cat text.1; cat text.2} | lpr - пpосмотp файлов text.1 и text.2 и вывод их

на печать;  
ps [-al] [number] - команда для вывода информации о процессах:  
-a - вывод информации обо всех активных процессах, запущенных с

вашего терминала;  
-l - полная информация о процессах;  
number - номер процесса.  
Команда ps без параметров выводит информацию только об активных

процессах, запущенных с данного терминала, в том числе и фоновых. На экран выводится подробная информация обо всех активных процессах в следующей форме:

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD

1 S 200 210 7 0 2 20 80 30 703a 03 0:07 cc 1 R 12 419 7 11 5 20 56 20 03 0:12 ps

F - флаг процесса (1 - в оперативной памяти, 2 - системный процесс, 4 - заблокирован в ОЗУ, 20 - находится под управлением другого процесса, 10 - подвергнут свопингу);

S - состояние процесса (O - выполняется процессором , S - задержан, R

- готов к выполнению, I - создается);  
UID - идентификатор пользователя;  
PID - идентификатор процесса;  
PPID - номер родительского процесса;  
C - степень загруженности процессора;  
PRI - приоритет процесса, вычисляется по значению переменной NICE

и чем больше число, тем меньше его приоритет;  
NI - значение переменной NICE для вычисления динамического

приоритета, принимает величины от 0 до 39;  
ADDR - адрес процесса в памяти;  
SZ - объем ОЗУ, занимаемый процессом;  
WCHAN - имя события, до которого процесс задержан, для активного

процесса - пробел;  
TTY - номер управляющего терминала для процесса;  
TIME - время выполнения процесса;  
CMD - команда, которая породила процесс.  
nice [-приращение приоритета] команда[аргументы] - команда

изменения приоритета. Каждое запущенное задание (процесс) имеет номер приоритета в диапазоне от 0 до 39, на основе которого ядро вычисляет фактический приоритет, используемый для планирования процесса. Значение 0 представляет наивысший приоритет, а 39 - самый низший. Увеличение номера приоритета приводит к понижению приоритета, присвоенного процессу. Команда nice -10 ls -l увеличивает номер приоритета, присвоенный процессу ls -l на 10.

renice 5 1836 - команда устанавливает значение номера приоритета процесса с идентификатором 1836 равным 5. Увеличить приоритет процесса может только администратор системы.

kill [-sig] <идентификатор процесса> - прекращение процесса до его программного завершения. Sig - номер сигнала. Sig = -15 означает программное (нормальное) завершение процесса, номер сигнала = -9 - уничтожение процесса. По умолчанию sig= -9. Вывести себя из системы можно командой kill -9 0. Пользователь с низким приоритетом может прервать процессы, связанные только с его терминалом.

mc - вызов файлового менеджера (программы - оболочки) Midnight Commander, аналогичного Norton Commander.

sort [-dr] - сортировка входных файлов и вывод результата на экран.

4. Краткое описание командного интерпретатора Shell

Интерпретатор команд Shell анализирует команды, вводимые с терминала либо из командного файла, и передает их для выполнения в ядро системы. Команды обычно имеют аргументы и параметры, которые обеспечивают модернизацию выполняемых действий. Shell является также языком программирования, на котором можно создавать командные файлы (shell-файлы). При входе в ОС пользователь получает копию интерпретатора Shell в качестве родительского процесса. Далее, после ввода команды

пользователем создается порожденный процесс, называемый процессом- потомком. Т.е. после запуска ОС каждый новый процесс функционирует только как процесс - потомок уже существующего процесса. В ОС Linux имеется возможность динамического порождения и управления процессами.

Обязательным в системе является интерпретатор Bash, полностью соответствующий стандарту POSIX. В качестве Shell может быть использована оболочка mc с интерфейсом, подобным Norton Commander.

3. Задания для лабораторной работы

Изучить работу в командном интерпретаторе BASH.

4. Методика выполнения задания

1. Ознакомиться с теоретической частью к лабораторной работе.  
2. Определить день недели, в который Вы родились.  
3. Получить подробную информацию обо всех активных процессах.  
4. Используя редактор VI (см. приложение), создать два текстовых

файла (с расширением TXT) и командой САТ просмотреть их на экране.  
5. Получить информацию о работающих пользователях, подсчитать их

количество и запомнить в файле.  
6. Объединить текстовые файлы в единый файл и посмотреть его на

экране.  
7. Посмотреть приоритет своего процесса и уменьшить скорость его

выполнение за счет повышения номера приоритета.  
8. Используя редактор VI, написать программу на языке Си и запустить

ее на трансляцию в фоновом режиме.  
9. Показать преподавателю исходный текст программы на языке Си,

текстовый файл, файл с сохранением количества пользователей. 10. Продемонстрировать выполнение Си - программы.  
11. Удалить свои файлы и выйти из системы.

5. Требования к содержанию и оформлению отчета

Отчет по лабораторной работе должен содержать:  
а) титульный лист;  
б) описание хода выполнения работы и снимки экрана; в) заключение по выполненной работе;  
г) ответы на контрольные вопросы

Контрольные вопросы

1. Перечислите основные функции и назначение многопользовательской многозадачной операционной системы LINUX и ее отличительные особенности от однопрограммной системы DOS.

2. Какое назначение имеет ядро системы и интерпретатор команд?

3. В чем заключается понятие «процесс» и какие операции можно выполнить над процессами?