Лабораторная работа 02

OC, ИСиТ-3

**Задание 01. Команды (утилиты) Linux**

1. Исследуйте назначение следующих стандартных утилит Linux
2. Составьте таблицу поясняющую назначение команд (п.1).

|  |  |
| --- | --- |
| Команда Linux | Краткое описание команды |
| ls | вывод списка файлов и каталогов |
| echo | выводит строку текста в терминал |
| pwd | печатает на экран текущий каталог |
| cd | позволяет перейти из текущего каталога в указанный |
| mkdir | создание новых каталогов |
| rmdir | Удаление каталогов |
| touch | установка времени последнего изменения файла или доступа в текущее время, создание файла |
| cp | копирование файлов и каталогов -r (**R**ecursive) или -a (**A**rchive) |
| mv | перемещение или переименование файлов и каталогов |
| rm | удаляет файлы и папки |
| su | запустить программу от имени другого пользователя |
| whoami | отображает имя вошедшего в систему пользователя |
| man | открывает руководство по определённой команде |
| whereis | показывает полный путь к исполняемому файлу программы. Также может показать путь к исходникам, если они есть в системе. |
| whatis | показывает, какие разделы руководств есть для данной команды |
| apropos | осуществляет поиск переданной пользователем строки в заголовках страниц руководств |
| cat | печатает содержимое файла, переданного в параметре, в стандартный вывод |
| less | для просмотра длинных текстов, которые не вмещаются на одном экране |
| head | выводит несколько первых строк из файла (голова),(10 строк по умолчанию) |
| tail | выдает несколько последних строк (хвост). |
| lpstat | вывод информации о состоянии системы печати |
| lpr | поставить файл в очередь на печать |
| lpq | просмотреть очередь на печать |
| lprm | удалить запрос из очереди |
| chgrp | позволяет задействовать соответствующую утилиту для изменения группы пользователей, владеющих файлом или директорией |
| chown | изменяет владельца файла (только суперпользователь может изменять владельцев) |
| chmod | Изменяет права доступа к файлу. Это чтение, запись и выполнение. Каждый пользователь может изменять права для своих файлов. |
| zip | создание zip-архива |
| gzip | сжатия данных без потерь |
| gunzip | восстановление сжатых файлов |
| bzip2 | предназначена для сжатия данных без потерь с помощью соответствующей утилиты, использующей [алгоритм Барроуза-Уилера](https://ru.wikipedia.org/wiki/Bzip2) |
| bunzip2 | символьной ссылкой на утилиту bzip2 с параметром -d |
| tar | cоздать новый архив |
| locate | ведёт поиск в базе данных updatedb для шаблонов имён файлов. Эта база данных содержит снимок файловой системы, что позволяет искать очень быстро |
| grep | ищет текст по шаблону |
| find | поиск в файловой системе, файлах и папках |
| history | команда для повторного исполнения ранее исполненных команд |
| alias | Команда создаёт синонимы для других команд Linux. То есть вы можете делать новые команды или группы команд, а также переименовывать существующие |
| unalias | удаление |
| ps | выводит список идентификаторов текущих процессов на вашем сервере |
| top | отображают процессы и могут быть использованы как консольные системные мониторы |
| lsof | для отображения открытых файлов различными процессами и/или пользователями |
| free | предоставляет информацию об использованной и неиспользованной памяти, а так же о разделе подкачки (swap) |
| df | анализатор дискового пространства |
| du | показывает размер файла или каталога |
| yum | средство обновления пакетов |
| ifconfig | возвратит список подключенных к системе сетевых интерфейсов и их характеристики, например, ip адрес, адрес шлюза, размер пакета, частоту для беспроводных сетей и другие параметр |
| ping | в основном используется для проверки доступности удаленного хоста или нет |
| traceroute | Это усовершенствованная версия ping. Мы можем увидеть не только полный маршрут сетевых пакетов, но и доступность узла, а также время доставки этих пакетов на каждый из узлов. |
| host | предназначена для выполнения запросов к DNS-серверам. По умолчанию она просто находит IP-адрес, соответствующий заданному имени хоста |
| iwconfig | настроить беспроводной сетевой интерфейс |
| dhclient | для управления адресом интерфейса по протоколу DHCP |
| ifup | включения сетевого интерфейса |
| ifdown | выключения сетевого интерфейса |
| route | посмотреть таблицу маршрутизации |
| ssh | соединяет вас с вашим сервером, который имеет свой IP адрес serverip и имя пользователя user.( ssh user@serverip) |
| sftp | открыть SFTP-сессию через это соединение |
| scp | позволяет выполнить безопасное копирование и перенос файлов |
| rsync | эффективно передаёт и синхронизирует файлы или каталоги между локальной и удалённой машиной, удалённой оболочкой и соответствующей ей парой, другим хостом |
| wget | текстовая программа для скачивания файлов |
| curl | инструмент для передачи данных с сервера или на него |

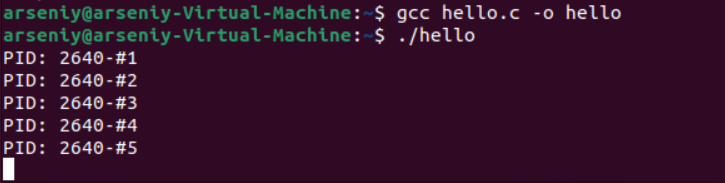
**Задание 02.Переменные окружения среды Linux**

1. Исследуйте назначение следующих стандартных переменных окружения
2. Составьте таблицу поясняющую назначение переменных окружения (п.3).

|  |  |
| --- | --- |
| Переменная окружения | Краткое описание |
| $HOME | здесь содержатся домашние каталоги всех пользователей, которые зарегистрированы в системе |
| $PATH | представляющая собой набор каталогов, в которых расположены исполняемые файлы |
| $PS1 | строка приглашения на ввод(Приглашение ко вводу команд в bash — это строка, которая отображается слева от каждой команды, которую вы вводите в терминале) |
| $PS2 | используется, когда команда многострочная |
| $IFS | внутренний разделитель полей для разделения ввода в командной строке |

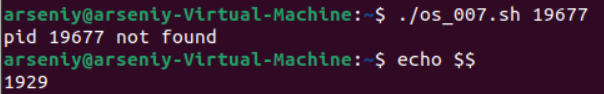
**Задание 04.Разработка С-приложения и bash-скрипт**

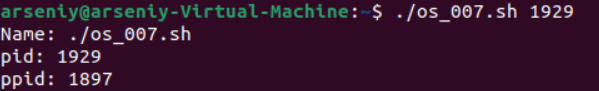
1. Разработал приложение на языке С, выводящее 1000 сообщений на консоль. Сообщения выводиться каждые 2 секунды на консоль и содержат PID процесса и номер сообщения.

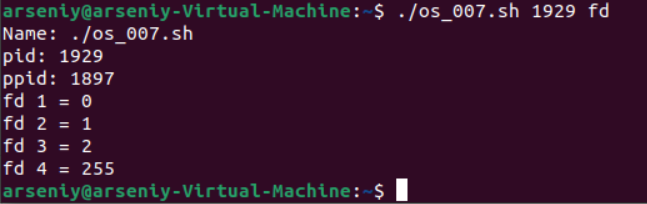


|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <errno.h>  #include <sys/types.h>  #include <sys/wait.h>  int main()  {  for (int a = 1; a <= 1000; a++)  {  sleep(2);  printf("PID: %d-#%d\n", getpid(), a);  }  exit(0);  } |

1. Разработайте bash-скрипт **os\_0007.sh**, который принимает два параметра: Pid процесса, строку fd (необязательный параметр). В результате работы bash-скрипт выводит в консоль: наименование исполняемого файла, Pid-процесса, Pid-родительского процесса (PPid), перечень дескрипторов (номеров) дескрипторов (fd) открытых потоков. Приведенные ниже скриншоты демонстрируют работу скрипта os\_0007.sh.



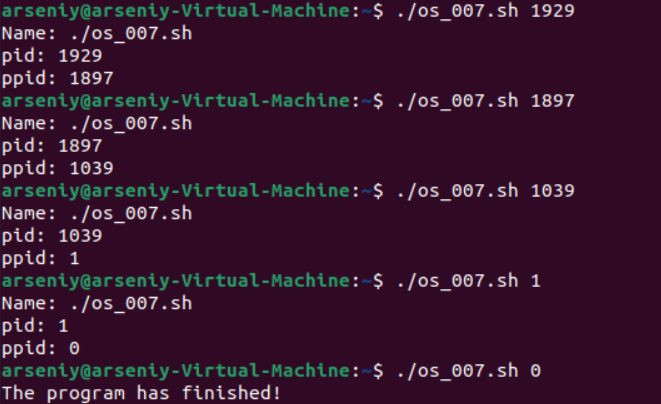




|  |
| --- |
| #!/bin/bash  if [ "$1" -eq "0" ]; then  echo "The program has finished!"  exit  fi  if [ -z "$1" ]; then  echo "parameters not found"  exit  fi  if ! [[ $1 =~ ^[+-]?[0-9]+$ ]]; then  echo "$1 is not an integer"  exit  fi  if ps -p $1 > /dev/null; then  echo "Name: $0"  echo "pid: $1"  echo "ppid: $(ps -o ppid= -p $1 | tr -d ' ')"    if [ "$2" == "fd" ]; then  line=1  for variable in $(ls -1 /proc/$1/fd); do  echo "fd $line = $variable"  ((line++))  done  fi  else  echo "pid $1 not found"  exit  fi |

**Задание 05.Построение цепочки родительских процессов**

1. Применив, разработанный скрипт **os\_0007.sh**  постройте цепочку родительских процессов до процесса инициализации system.Ниже приводится пример, первых трех шагов построения цепочки.



**Задание 06.ответьте на следующие вопросы**

1. Что такое фреймворк OS?
2. Что такое POSIX?
3. Что такое аппаратное прерывание?
4. Что такое программное прерывание?
5. Что такое системный вызов?
6. Что такое процесс OS?
7. Что такое контекст процесса OS?
8. Что такое адресное пространство процесса?
9. Перечислите области памяти адресного пространства процесса и поясните их назначение.
10. Что такое стандартные потоки процесса?
11. Перечислите системные вызовы Windows для создания процесса?
12. Перечислите системные вызовы Linux для создания процесса?
13. С помощью каких утилит можно увидеть перечень процессов в Windows?
14. С помощью каких утилит можно увидеть перечень процессов в Linux?
15. Перечислите свойств а процесса OS.
16. Что такое фреймворк OS?

[Программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), облегчающее разработку и объединение разных компонентов большого программного проекта.

1. Что такое POSIX?

POSIX (англ. Portable Operating System Interface — переносимый интерфейс операционных систем) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов.

1. Что такое аппаратное прерывание?

**Прерывание** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *interrupt*) — сигнал от программного или аппаратного обеспечения, сообщающий [процессору](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) о наступлении какого-либо события, требующего немедленного внимания.

асинхронные, или внешние (аппаратные) — события, которые исходят от внешних аппаратных устройств (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши.

Аппаратное - реакция процессора на физический сигнал от некоторого устройства (клавиатура, системные часы, клавиатура, жесткий диск и т.д.), по времени возникновения эти прерывания асинхронны, т.е. происходят в случайные моменты времени;

1. Что такое программное прерывание?

программные (частный случай внутреннего прерывания) — инициируются исполнением специальной [инструкции](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8) в коде [программы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0).

 Программные - вызываются искусственно с помощью соответствующей команды из программы, предназначены для выполнения некоторых действий операционной системы, являются синхронными;

1. Что такое системный вызов?

Системный вызов - механизм вызова прикладной программой функции ядра OS.

обращение прикладной программы к ядру операционной системы для выполнения какой-либо операции.

1. Что такое процесс OS?

Процесс OS – единица работы OS - объект ядра OS+адресное пространство:

Процесс OS – это выполняемая в данный момент программа

1. Что такое контекст процесса OS?

Контекст процесса – данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы; (адресное пространство, содержимое регистров, объекты ядра OS (объекты процессов, потоков, безопасности, файлов и пр.) стек ядра (для этого процесса))

1. Что такое адресное пространство процесса?

адресное пространство (данные, программа, стек, куча);

Адресное пространство — это просто диапазон адресов, обозначающих определенное место в памяти. Адресные пространства подразделяются на три разновидности:

* Физическое адресное пространство
* Линейное адресное пространство
* Логическое адресное пространство, известное также как виртуальное адресное пространство

Физические адреса — это реальные, аппаратные адреса, доступные в системе. Если в системе имеется 64 Мб памяти, в ней допустимые физические адреса могут находиться в диапазоне от 0 до 0x3fffffff (в шестнадцатиричном формате). Каждый адрес соответствует одному набору транзисторов в микросхемах SIMM, установленных вами (или изготовителем), и отдельному сочетанию сигналов на адресной шине процессора.

Страничный обмен позволяет перемещать процессы или только фрагменты процессов в различные области физической памяти (различные физические адреса) и обратно в течение срока существования процесса. Именно по этой причине процессам предоставляется пространство логических адресов

1. Перечислите области памяти адресного пространства процесса и поясните их назначение.

процессу выделяется линейное адресное пространство (размер зависит от разрядности), сегменты: code, static, data, heap, stack;



1. Что такое стандартные потоки процесса?

потоки имеющие зарезервированные, для выполнения некоторых “стандартных” функций, номера - дескрипторы (номера), поток ввода (0), поток вывода (1), поток вывода ошибок (2).

1. Перечислите системные вызовы Windows для создания процесса?

CreateProcess, которая создает новый процесс с единственным потоком. При вызове этой функции требуется указать имя файла исполняемой программы.

CreateProcess, CreateProcessAsUser, CreateProcessWithTokenW, CreateProcessWithLogonW

1. Перечислите системные вызовы Linux для создания процесса?

Fork(),exec()

1. С помощью каких утилит можно увидеть перечень процессов в Windows?

tasklist - показывает список процессов на локальном или удаленном компьютере

taskkill - позволяет завершить  процессы

1. С помощью каких утилит можно увидеть перечень процессов в Linux?

ps - показывает запущенные процессы

top - выводит список работающих в системе процессов и инф о них

lsof - List Of Opened Files - отображения открытых файлов различными процессами и/или пользователями

1. Перечислите свойства процесса OS.

OS: основные свойства процесса:

* процессу соответствует исполняемый программный файл;
* у процесса есть PID;
* у процесса есть Parent PID;
* в Windows: HANDLE – идентификатор объекта OS;
* в OS есть процесс инициализации (родитель для всех);
* запуск и управление (создать, остановить,…) процессом осуществляется с помощью системных вызовов;
* процессы изолированы друг от друга;
* процессу выделяется линейное адресное пространство (размер зависит от разрядности), сегменты: code, static, data, heap, stack;
* контекст процесса – данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы;
* процессу автоматически доступны три процесса: ввода, вывода, вывод ошибок.
* при запуске OS некоторые процессы (Windows-сервисы, Linux-демоны) загружаются и стартуют автоматически, как правило используются для внутреннего назначения;
* в составе ОS есть таблица, содержащая объекты ядра процессов (состояние, приоритет, указатели на другие объекты); есть средства OS позволяющие ее просматривать;
* процесс – единица работы OS.