Лабораторная работа 02

OC, ПОИТ-3

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

1. Исследуйте назначение следующих стандартных утилит Linux:

echo,

ls, pwd, cd,

mkdir, rmdir,

touch,

cp, mv, rm

su, whoami,

man, whereis, whatis, apropos,

cat, less, head, tail,

lpstat, lpr, lpq, lpstat, lprm,

chgrp, chown, chmod,

zip, gzip, gunzip, bzip2, bunzip2, tar,

locate, grep, find,

history,

alias, unalias,

ps, top, lsof,

free, df, du,

yum,

ifconfig, ping, traceroute, host, iwconfig, dhclient, ifup, , route,

ssh, sftp, scp, rsync, wget, curl,

1. Составьте таблицу поясняющую назначение команд (п.1).

|  |  |
| --- | --- |
| Команда Linux | Краткое описание команды |
| **ls** | вывод списка файлов и каталогов… |
| … | … |
| **curl** | Копирование файлов web-узла… |

**Задание 02.Переменные окружения среды Linux**

1. Исследуйте назначение следующих стандартных переменных окружения

$HOME

$PATH

$PS1

$PS2

$IFS

1. Составьте таблицу поясняющую назначение переменных окружения (п.3).

|  |  |
| --- | --- |
| Переменная окружения | Краткое описание |
| $HOME | исходный каталог … |
| … | … |
| $IFS | разделитель … |

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

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



* 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.









**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?- набор функций для работы с ядром ос

Что такое POSIX? – набор соглашений и правил для UNIX под xобных систем. описывающих интерфейсы между

1. операционной системой и прикладной программой
2. Что такое аппаратное прерывание? – это когда процессор прерывает работу менее приорететного устройства пк для работы более приорететного
3. Что такое программное прерывание? – когда прерываение происходит в прикладной программе
4. Что такое системный вызов? Это обращение к ядру ОС для выполенение какой либо операции
5. Что такое процесс OS? Это объект ядра ОС который скрывает исполняющийся код от других процессов
6. Что такое контекст процесса OS? METADATA или же инфа о процессе  
   Контекст процесса в операционной системе (OS) - это состояние процесса, которое включает информацию о его текущем выполнении
7. Что такое адресное пространство процесса?

Адресное пространство процесса - это набор адресов, которые процесс может использовать для доступа к памяти.

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

- Сегмент кода (Code segment): Эта область содержит исполняемый код программы.

- Сегмент данных (Data segment): Здесь хранятся глобальные и статические переменные программы.

- Сегмент стека (Stack segment): В этой области размещаются локальные переменные и данные, относящиеся к текущим вызовам функций.

- Сегмент кучи (Heap segment): Здесь располагаются динамически выделяемые данные, такие как объекты, массивы и структуры данных.

1. Что такое стандартные потоки процесса?

**Поток (или нить выполнения)** - это последовательность инструкций, которые выполняются в рамках процесса. Потоки являются более легковесными, чем процессы, поскольку они используют общие ресурсы процесса, такие как память и файловые дескрипторы. Внутри одного процесса может быть создано несколько потоков, которые работают параллельно или конкурируют за ресурсы процесса. Каждый поток имеет свой собственный стек выполнения, указатель команд и набор регистров.

Стандартные потоки процесса - это три основных потока ввода-вывода, связанных с каждым процессом: стандартный ввод (stdin), стандартный вывод (stdout) и стандартный вывод ошибок (stderr). Они представляют собой каналы, через которые процесс взаимодействует с окружающей средой. Например, стандартный ввод может использоваться для чтения пользовательского ввода, а стандартный вывод и вывод ошибок - для вывода результатов и сообще

1. Перечислите системные вызовы Windows для создания процесса?

Некоторые системные вызовы Windows для создания процесса:

- CreateProcess: Создает новый процесс и его первичный поток выполнения.

- CreateProcessAsUser: Создает новый процесс от имени указанного пользователя.

- CreateProcessWithLogonW: Создает новый процесс с использованием имени пользователя и пароля для аутентификации.

1. Перечислите системные вызовы Linux для создания процесса?

Некоторые системные вызовы Linux для создания процесса:

- fork: Создает новый процесс путем дублирования вызывающего процесса. Новый процесс называется "потомком", а исходный процесс - "родителем".

- exec: Заменяет текущий процесс новым процессом, загружая и выполняя новую программу из исполняемого файла.

- clone: Создает новый поток или процесс с заданными параметрами. Он предоставляет более гибкий контроль над создаваемыми процессами и потоками

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

- Диспетчер задач (Task Manager): Встроенная утилита, которая позволяет просматривать и управлять запущенными процессами, мониторить системные ресурсы и многое другое.

- PowerShell: Командлеты Get-Process и Tasklist позволяют получить информацию о запущенных процессах.

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

В Linux для просмотра перечня процессов можно использовать следующие утилиты:

- ps: Команда ps отображает текущие процессы в виде списка.

- top: Утилита top позволяет просматривать список процессов в реальном времени и мониторить использование системных ресурсов.

- htop: Альтернативная утилита для просмотра процессов с интерактивным интерфейсом.

1. Перечислите свойства процесса OS.

Свойства процесса в операционной системе могут включать:

- Идентификатор процесса (PID): Уникальный числовой идентификатор, который идентифицирует процесс в системе.

- Состояние процесса: Текущее состояние процесса, например, запущен, приостановлен, ожидает и т. д.

- Приоритет процесса: Значение, определяющее относительную важность процесса и его приоритет выполнения.

- Ресурсы процесса: Информация о выделенных ресурсах, таких как память, файловые дескрипторы, открытые файлы и сетевые соединения.

- Родительский процесс(ppid): Информация о процессе, который создал текущий процесс (родительский процесс).

- Потоки процесса: Список потоков выполнения, работающих в рамках процесса.

- Время выполнения: Информация о времени, затраченном процессом на выполнение задач.

- Используемые системные ресурсы: Информация о системных ресурсах, используемых процессом, таких как процессорное время, память и дисковое пространство.