

2 Управление процессами

2.1 Цель работы

2.1.1 Изучить работу процессов средствами Windows

2.1.2 Изучить основные процессы, запускаемые ОС Linux

2.2 Приборы и оборудование

2.2.1 ПЭВМ типа IBM PC

2.2.2 ОС Linux

2.3 Порядок выполнения работы

2.3.1. Изучение работы процессов средствами диспетчера задач Windows

- a) Загрузите виртуальную машину Windows XP (WXPadm)
- b) Для запуска диспетчера задач и просмотра компонентов вычислительного процесса нужно выполнить следующие действия: щелкнуть кнопкой мыши по панели задач и выбрать строку **Диспетчер задач**.
- c) Для просмотра приложений перейти на вкладку **Приложения**. Здесь можно завершить приложение (кнопка **Снять задачу**), переключиться на другое приложение (кнопка **Переключится**) и создать новую задачу (кнопка **Новая задача**). В последнем случае после нажатия кнопки **Новая задача** появится окно нужно ввести имя задачи.
- d) Выпишите в отчет список активных процессов, воспользовавшись вкладкой **Процессы**.
- e) Попробуйте завершить процесс *winlogon.exe*. Какова реакция системы?
- f) Запустите редактор *WordPad*. С помощью **Диспетчера задач**, определите какой процесс связан с приложением *Документ-WordPad*.
- g) В окне **Диспетчера задач** перейдите на вкладку **Приложения**. В контекстном меню приложения *Документ-WordPad* (вызывается щелчком левой кнопкой мышки по имени приложения в списке) выполните **Перейти к процессам**
- h) В окне **Диспетчера задач** перейдите на вкладку **Процессы**. Какой процесс инициализируется при запуске *WordPad*. Запишите его идентификатор (**PID**) (если данный столбец отсутствует на страницу процессов диспетчера задач, то его можно добавить, воспользовавшись пунктом меню Вид/Выбрать столбцы). Запустите еще один экземпляр редактора *WordPad*. Определите идентификатор связанного с ним процесса. Отметьте полученный результат в отчете. Ответьте на вопрос, создается ли новый процесс при запуске нового экземпляра приложения *WordPad*?

- i) С помощью **Диспетчера задач**, определите, сколько потоков создается процессом *WordPad.exe*. В окне **Диспетчера задач** перейдите на вкладку **Процессы**. В меню **Вид** выполните команду **Выбрать столбцы**, затем включите **Счетчик потоков** и нажмите **Ок**. Столбец **Потоков** на вкладке **Процессы** и содержит количество потоков, создаваемых соответствующим процессом. Отметьте в отчете, сколько потоков создает процесс *WordPad.exe*
- j) С помощью **Диспетчера задач** завершите работу приложений *WordPad*
- к) С помощью **Диспетчера задач**, определите:
- создается ли новый процесс при запуске нового экземпляра приложения *Калькулятор*;
 - сколько при этом создается потоков;
 - определите процесс, имеющий максимальный идентификационный номер зафиксируйте имя процесса и его PID;
 - запустите приложение *Paint*. Определите, какой идентификационный номер присвоила операционная система соответствующему процессу. Запишите полученный результат в отчет.
- l) Запустите системный монитор Windows. В меню **Пуск** выполните **Настройка/Панель управления**. В окне **Панель управления** щелкните по **Администрирование/Производительность**, затем по **Системный монитор**.

Процесс	Total
% загрузки процессора	100,000
% работы в пользовательском режиме	3,000
% работы в привилегированном режиме	97,000
I/O - запись байт в сек	0,000
I/O - обмен данными, байт в сек	0,000
I/O - операций записи в сек	0,000
I/O - операций с данными в сек	0,000
I/O - операций чтения в сек	0,000
I/O - прочий байт в сек	151,784
I/O - прочий операций в сек	4,993
I/O - чтение байт в сек	0,000
Базовый приоритет	0,000
Байт в выгружаемой страничном пуле	1229768,000
Байт в невыгружаемой страничном пуле	1552384,000
Байт виртуальной памяти	1512439808
Байт виртуальной памяти (пик)	1801797632
Байт исключительного пользования	153190400,000
Байт файла подкачки	153190400,000
Байт файла подкачки (пик)	187576320,000
Идентификатор процесса	0,000
Код (PID) создавшего процесса	n nnn

- m) В окне **Системного монитора**. Включите режим **Просмотр отчета**, щелкнув по одноименной кнопке на панели инструментов **Системного монитора**.
- n) Используя **Системный монитор** определите количество активных в текущий момент процессов и потоков
- o) Выведите в окне **Системного монитора** для всех активных процессов значения идентификатора процесса и код (ID) создавшего процесса

- p) Запустите редактор WordPad через меню **Пуск/Программы**
- q) Щелкните по инструменту **Добавить**
- r) В окне **Добавить счетчики** Включите **использовать локальные счетчики**. В списке **Объект** выберите **Процесс**. В списке **Выбрать счетчики из списка** выберите счетчики **Идентификатор процесса** и **текущий приоритет**. Выберите **Все вхождения**
- s) Выполните **Добавить**, затем **Заккрыть**
- t) Заполните и поместите в отчет следующую таблицу

Процесс	Идентификатор процесса	Код (ID) и имя создавшего процесса-родителя

- u) Завершите сеанс работы Windows

2.3.2. Изучение работы процессов средствами ОС Linux

- a) Загрузите ОС Linux
- b) Запустите утилиту **Системный монитор** (Меню запуска приложений/Система/Системный монитор(KSysGuard)).
- c) Просмотрите запущенные процессы. Выберите любой созданный процесс и опишите его по следующей схеме:
 - количество нитей исполнения;
 - PID процесса;
 - родительский процесс;
 - ID родительского процесса;
 - общее время работы процесса;
 - объем виртуальной памяти, используемой процессом.
- d) Отфильтруйте процессы, оставив «только системные». Дайте характеристику одному системному процессу (выбор процессов осуществляется на панели Таблица процессов в списке) по следующей схеме:
 - идентификатор процесса и его название
 - идентификатор пользователя, запустившего процесс;
 - приоритет процесса;
 - размер виртуальной и физической памяти, занимаемой процессом;
 - используя идентификатор породившего процесса, определите какой процесс являлся процессом-предком этого процесса (посмотрите в сплывающем меню процесса ID родительского процесса, найдите это число в PID и, тем самым определите процесс, который является процессом-предком данного процесса).

- e) Отфильтруйте процессы, оставив «только пользовательские». Опишите по вышепредложенной схеме один пользовательский процесс. С помощью меню, вызванного нажатием правой кнопки мыши, отправьте процессу сигнал Приостановить (STOP). Какие изменения произошли? Какие еще сигналы можно отправить процессу?
- f) Запустите программу Терминал (Меню запуска приложений\Система\Терминал(Konsole)). Какой процесс был порожден, и каков его приоритет. Имя процесса – konsole.

Выведите информацию обо всех выполняющихся процессах с помощью команды **ps aux**:

Оставьте в выводе предыдущей команды только системные процессы с помощью команды **ps aux | grep -v user** (часть процессов, выведенные в квадратных скобках, — это потоки ядра, исполняющие специальные функции. Можно выделить демонов — в столбце терминала они имеют символ «?»). Например, системный журнал (**syslog**) или планировщик задач (**cron**)

- g) Выведите иерархию процессов с помощью команды **pstree**. Какой процесс стоит в вершине иерархии?
- h) Рассмотрите поведение процессов интерактивно с помощью команды **top** (выход из просмотра клавиша Q).
- i) Отобразите все существующие в системе процессы командой **ps aux**. Приведите листинг команды и расшифруйте значение столбцов (**man ps** или воспользуйтесь теоретическими сведениями данной лабораторной работы).

2.3.21. Завершите сеанс работы с Linux.

2.3.22. Оформите отчет и сдайте его преподавателю.

2.4 Контрольные вопросы

2.4.1. Дайте определение понятия «процесс».

2.4.2. Что означает термин «поток»?

2.4.3. Дайте определение понятию «приоритет».

2.4.4. Для чего предназначена программа «Системный монитор»?

2.4.5. Какие характеристики процесса относят к дескриптору процесса?

2.4.6. Перечислите изученные вами команды работы с процессами.

2.4.7. Какие способы запуска Диспетчера процессов вы знаете?

Для управления процессами существует такая подсистема как **менеджер процессов**. В **Linux** реализована многозадачность с вытеснением и менеджер процессов полностью управляет процессами. Каждый процесс имеет свой контекст. Контекст процесса это таблица распределения памяти, текущий статус процесса, приоритет процесса, данные о используемых ресурсах, маска сигналов, идентификаторы процесса и другие свойства. Идентификатор процесса состоит из идентификатора **PID** и идентификатора родительского процесса **PPID (ID)**.

Самый первый пользовательский процесс называется **init**. Его **PID** всегда равен 1, а **PPID (ID)** равен 0, так как он не имеет родительского процесса, а порождается ядром на определенном этапе загрузки системы. Все остальные процессы являются дочерними процессами и имеют **PPID (ID)** отличный от нуля. Для того чтобы породить новый процесс, родительский вызывает функцию ядра **fork()**, в результате которой происходит клонирование родительского процесса и получается абсолютно одинаковый процесс на начальном этапе. Единственное отличие: у дочернего процесса увеличивается **PID**. **PID** дочернего процесса всегда будет больше чем у родительского. В остальном контекст полностью будет совпадать с родительским. Затем для дочернего процесса запускается функция **exec()**, которая начинает уже изменять контекст на нужный, загружать в память код программы и т.д.

В определенный момент времени работать может только один процесс. Но так как переключение между процессами происходит очень быстро (тысячи переключений в секунду), то создается впечатление, что все они работают одновременно. Сколько процессорного времени и других ресурсов выделить каждому процессу решает менеджер процессов. В принятии этого решения играет роль такой параметр процесса как **nice** (или **nice-фактор**). **nice-фактор** - это числовое значение. Чем оно больше, тем меньше ресурсов (процессорного времени и т.д.) будет выделяться процессу во время его работы.

Многие процессы выполняют операции ввода/вывода информации в/из терминала. Поэтому есть такой параметр как **Терминал**. А есть процессы, которые работают в фоновом режиме. Такие процессы не привязаны к терминалу. Например, процесс **cron**.

Для просмотра процессов и их свойств предназначена команда **ps**. Команда имеет очень много ключей и параметров. Все их можно посмотреть в **man ps**. Программе **ps** можно передавать как параметры, так и ключи. Напомню, что перед ключом (или последовательностью ключей) всегда стоит дефис, а параметры пишутся без дефиса. Сравните результаты команд **ps** с ключом (**ps -a**) и с параметром (**ps a**). Как видите, результаты существенно отличаются. А теперь рассмотрим еще несколько распространенных вариантов команды **ps**.

Чтобы отобразить все существующие процессы предназначена команда **ps aux**.
Результат ниже сокращен (процессов гораздо больше):

```
igor@adm-ubuntu:~/linux$ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 11740 0.0 0.0 3124 448 ? S< 11:28 0:00 udevd --daemon
igor 11805 0.0 0.0 1848 548 pts/0 S+ 11:43 0:00 ping 192.168.222.10
igor 11915 0.0 0.1 2640 1040 pts/2 R+ 12:00 0:00 ps aux
```

По команде **ps aux** можно увидеть достаточно обширный объем информации.
Расшифровка столбцов следующая:

USER - владелец процесса

PID - **PID** процесса

%CPU - процент использования процессора. Это время использования процессора, деленное на общее время существования процесса.

%MEM - процент использования физической оперативной памяти.

VSZ - объем виртуальной памяти, выделенной процессу (в Килобайтах)

RSS - объем физической оперативной памяти, выделенной процессу (в Килобайтах)

TTY - к какому терминалу “привязан” процесс

STAT - статус (состояние) процесса. **R** - выполнение, **S** - ожидание, **T** - остановлен, **Z** - “зомби”

START - время когда был запущен процесс

TIME - количество процессорного времени, которое брал себе процесс. Обычно там нули или малое число (0:00 - 0:09). Если число большое, то процесс загружает процессор и следует проверить все ли с ним в порядке.

COMMAND - имя команды. Какой код был выполнен во время исполнения функции `exec()`.