

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управле-
ния»



Сёмкин П.С., Сёмкин А.П.

Методические материалы к лабораторным работам
по дисциплине
«Операционные системы»
(кафедра СГНЗ)

Лабораторная работа № 9
«OS Alt Linux. Управление процессами»

**Москва
2024 г.**

ОГЛАВЛЕНИЕ

1	ЦЕЛЬ РАБОТЫ	3
2	ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	3
2.1	Подсистема управление процессами	3
2.2	Организация процессов и потоков	3
2.2.1	Структуры данных процессов и потоков	3
2.2.2	Граф состояния задачи	3
2.3	Планирование и диспетчеризация в ОС Alt Linux	4
2.3.1	Алгоритм диспетчеризации	4
2.3.2	Приоритет планирования	4
2.3.3	Очередь выполнения	5
2.3.4	Изменение приоритета при планировании	5
2.3.5	Операции планирования	6
2.3.6	Многопроцессорное планирование	6
2.3.7	Планирование реального времени	7
2.4	Управление процессами	7
2.4.1	Монитор процессов Htop	8
2.4.2	Монитор процессов «Системный монитор»	8
2.4.3	Утилиты командной строки для управления процессами	8
3	ВЫПОЛНЕНИЕ РАБОТЫ	9
3.1	Задание	9
3.2	Порядок выполнения работы	9
3.2.1	Выполнить мониторинг и управление процессами с помощью монитора процессов htop 9	9
3.2.2	Выполнить мониторинг и управление процессами с помощью графической утилиты «Системный монитор»	10
3.2.3	Выполнить мониторинг и управление процессами с помощью утилит командной строки 10	10
4	КОНТРОЛЬНЫЕ ВОПРОСЫ	11
5	ЛИТЕРАТУРА	11
6	ПРИЛОЖЕНИЕ	12
6.1	Монитор процессов htop	12
6.1.1	Запуск монитора	12
6.1.2	Область с общей информацией о системе	12
6.1.3	Область с подробной информацией о процессах	12
6.1.4	Назначение функциональных клавиш	13
6.1.5	Клавиши для перемещения по окну программы	14
6.1.6	Поиск и фильтрация процессов	14
6.1.7	Изменение приоритета процесса	14
6.1.8	Выбор нескольких процессов	15
6.1.9	Закрытие процесса	15
6.1.10	Отображение только процессов определённого пользователя	15
6.1.11	Отображение только процессов с определёнными номерами	15
6.2	Утилита «Системный монитор»	15
6.2.1	Вкладка «Система»	15
6.2.2	Вкладка «Процессы»	16
6.2.3	Вкладка «Ресурсы»	16
6.2.4	Вкладка «Файловые системы»	16
6.3	Команды интерпретатора BASH мониторинга и управления процессами	17
6.3.1	Команды получения общей информации о системе	17
6.3.2	Команды получения информации о процессах	17
6.3.3	Команды управления процессами	18

1 Цель работы

Целью работы является знакомство и практическая работа со средствами мониторинга и управления процессами ОС Alt Linux.

2 Теоретическая часть

2.1 Подсистема управление процессами

Подсистема управления процессами предназначена для обеспечения эффективной реализации многозадачного режима в ОС Alt Linux.

Хотя в первую очередь она отвечает за **распределение процессоров** для выполнения процессов, функциями подсистемы управления процессами также являются **передача сигналов, загрузка модулей ядра и прием прерываний**.

В состав подсистемы управления процессами входит **планировщик процессов (process scheduler)**, обеспечивающий выделение процессорного времени процессам.

2.2 Организация процессов и потоков

2.2.1 Структуры данных процессов и потоков

В ОС семейства Linux и процессы, и потоки называют **задачами (task)**.

Изнутри они представляют собой единую структуру данных.

Каждая задача таблицы процессов представляется в виде структура **task_struct**, служащей в роли дескриптора процесса.

В структуре **task_struct** хранятся переменные и вложенные структуры, описывающие процесс.

2.2.2 Граф состояния задачи

Задача переходит в состояние **running (выполнения)** после выделения ей процессора.

При блокировке задача переходит в состояние **sleeping (спячки)**, а при остановке работы в состояние **останов (stopped)**.

Состояние **zombie (зомби)** показывает, что выполнение задачи прекратилось, однако она еще не была удалена из системы.

Задача в состоянии **dead (смерти)** может быть удалена из системы.

Состояния **active (активный)** и **expired (неактивный)** используются при планировании выполнения процесса.

При загрузке ядра запускается первый процесс **init**, который использует ядро для создания всех остальных задач.

2.3 Планирование и диспетчеризация в ОС Alt Linux

2.3.1 Алгоритм диспетчеризации

Диспетчеризация в ОС Alt Linux основана на потоках.

Алгоритм диспетчеризации различает три класса потоков:

1. **Потоки реального времени**, обслуживаемые по алгоритму FIFO.
2. **Потоки реального времени**, обслуживаемые в порядке циклической очереди.

3. Потоки разделения времени

Потоки реального времени, обслуживаемые по алгоритму FIFO, имеют наивысшие приоритеты и не могут прерываться другими потоками, за исключением такого же потока реального времени FIFO, перешедшего в состояние готовности.

Потоки реального времени, обслуживаемые в порядке циклической очереди, представляют собой то же самое, что и потоки реального времени FIFO, но с тем отличием, что они могут прерываться таймером.

Потоки разделения времени обслуживаются в режиме пакетной обработки с выделением им заданных квантов времени.

2.3.2 Приоритет планирования

У каждого потока есть **приоритет планирования**. Значение приоритета планирования по умолчанию равно **20**, но оно может быть изменено при помощи системного вызова **nice(value)**, вычитающего значение **value** из **20**. Поскольку **value** находится в диапазоне от **-20** до **+19** (Наивысший приоритет равен **-20**), приоритеты планирования всегда попадают в промежуток от **1** до **40**

Помимо приоритета с каждым потоком связан квант времени, то есть количество тиков таймера, в течение которых поток может выполняться. По

умолчанию системные часы тикают с частотой **100 Гц**, так что каждый тик равен **10 мс**. Этот интервал в ОС Alt Linux называют «**джиффи**» (jiffy - мгновение, миг, момент).

Центральный процессор отнимается у потока при выполнении одного из следующих условий:

1. **Квант потока уменьшился до 0.**
2. **Поток блокируется на операции ввода-вывода, семафоре и т. д.**
3. **В состоянии готовности перешел ранее заблокированный поток с более высокой «добродетелью».**

Так как кванты постоянно уменьшаются, рано или поздно у любого потока квант станет нулевым.

2.3.3 *Очередь выполнения*

После создания задачи с помощью команды `clone`, она помещается в **очередь выполнения** (run queue) процессора, содержащую ссылки на все задачи, состязющиеся за процессорное время. Очереди выполнения, напоминают многоуровневые очереди с обратной связью, что позволяет присваивать задачам различные приоритеты.

Когда задача переходит в состояние блокировки либо спячки (т.е. ожидания), или же ее выполнение прекращается по какой-либо иной причине, задача удаляется из очереди выполнения.

Чтобы отличить задачи, обладающие правом на процессорное время, от задач, которые вынуждены ожидать до наступления следующего периода дискретизации, в планировщике определены два состояния: **активный (active)** и **неактивный (expired)**. При этом планировщик осуществляет диспетчеризацию только тех задач, которые находятся в активном состоянии.

2.3.4 *Изменение приоритета при планировании*

В планировщике Alt Linux приоритет задачи влияет на размер кванта времени и порядок выполнения задач процессором.

Во время создания каждой задаче присваивается статический приоритет (**static priority**), называемый также правильным значением (**nice value**).

Планировщик различает 40 различных уровней приоритета: от -20 до 19.

Наименьшее значение означает наибольший приоритет в алгоритме планирования (т.е. -20 - это самый высокий приоритет, который может иметь процесс).

Одной из целей планировщика Alt Linux является обеспечение высокой степени интерактивности системы.

2.3.5 Операции планирования

Планировщик удаляет задачи из процессора в том случае, когда выполнение задачи прерывается, происходит ее приоритетное вытеснение (например, по окончании выделенного кванта времени) либо при блокировании задачи.

Каждый раз при удалении задачи из процессора, планировщик вычисляет для нее следующий квант времени.

В случае блокирования задачи либо невозможности ее выполнения по иной причине, она деактивируется (**deactivate**), то есть удаляется из очереди выполнения до тех пор, пока не будет снова готова к выполнению.

2.3.6 Многопроцессорное планирование

Поскольку планировщик процессов управляет задачами с помощью отдельных для каждого процессора очередей выполнения, задачи, как правило, являются структурно связанными с определенным процессором.

Это означает высокую вероятность отправки процесса в последующих периодах дискретизации на тот же самый процессор, что повышает быстродействие процесса, если его данные и инструкции все еще находятся в процессорном кэше.

Однако такая схема может привести к простоям одного или нескольких процессоров в многопроцессорной системе даже в то время, когда система испытывает серьезную нагрузку. Во избежание подобной ситуации в случае обнаружения простоя процесса, планировщик осуществляет **балансировку загрузки (load balancing)** для переноса задач с одного процессора на другой с целью повышения эффективности использования ресурсов.

Определение загрузки процессора выполняется планировщиком на основе данных о средней длине каждой очереди выполнения в течение нескольких последних прерываний таймера, чтобы минимизировать эффект непостоянства процессорной загрузки в алгоритме балансировки.

2.3.7 Планирование реального времени

Планировщик поддерживает жесткое планирование реального времени, пытаясь минимизировать время, затрачиваемое задачей реального времени на ожидание процессора.

В отличие от обычной задачи, которая, в конце концов, помещается в список неактивных (чтобы предотвратить бесконечное откладывание выполнения процессов с низким приоритетом), задача реального времени всегда помещается в активный список по истечению выделенного ей кванта времени.

Задачи реального времени всегда выполняются с более высоким приоритетом, чем обычные задачи. Поскольку планировщик всегда выделяет процессор задаче из очереди с самым высоким приоритетом активного списка (а задачи реального времени всегда находятся в активном списке), обычная задача никогда не сможет вытеснить задачу реального времени.

Чтобы предотвратить случайное либо злонамеренное использование задач реального времени, право на их создание имеют только пользователи с привилегиями **root**.

2.4 Управление процессами

Управление процессами ОС Alt Linux включает в себя выполнение следующих задач:

- *просмотр запущенных процессов*
- *просмотр информации о процессах*
- *поиск процессов*
- *изменение приоритета процессов*
- *завершение процессов*
- *ограничение памяти доступной процессу*

В Alt Linux есть множество утилит, как утилит командной строки, так и графических утилит, для решения различных задач по управлению процессами.

2.4.1 Монитор процессов Htop.

Монитор предназначен для просмотра в реальном времени информации о запущенных в системе процессах. Может быть запущен как из командной строки, так и из графической оболочки.

Htop показывает динамический список системных процессов, обычно выравненный по использованию ЦПУ. Показывается время непрерывной работы, использование процессоров и памяти.

Интерфейс программы приведён в приложении.

2.4.2 Монитор процессов «Системный монитор»

Системный монитор относится к графическим утилитам и позволяет просматривать список запущенных процессов, завершать процессы, следить за использованием памяти, центрального процессора и файловых систем.

Интерфейс утилиты приведён в приложении.

2.4.3 Утилиты командной строки для управления процессами

ОС Ubuntu, как и все ОС, содержит большой набор утилит командной строки для управления процессами

1. Утилита **ps**. Предназначена для мониторинга процессов в режиме реального времени. Показывает список всех процессов, которые выполнялись на момент запуска утилиты. Используется во многих случаях для вывода информации о выполняемых процессах. Может выполняться как в режиме командной строки, так и имеет графическую оболочку. Команда содержит множество параметров, наиболее часто используемые из которых приведены в приложении

2. Утилита **top**. Предназначена для вывода информации о процессах в реальном времени. Процессы сортируются по максимальному занимаемому процессорному времени, но есть возможность изменить порядок сортировки. Утилита также выводит информацию о свободных системных ресурсах.

3. Утилита **nice**. Позволяет определить текущего значения приоритета **nice** по умолчанию, понижать приоритет запускаемого или выполняемого процесса. Пользователь с правами **root** может повышать приоритет команды.

4. Утилита **kill**. Предназначена для принудительного завершения процесса.

5. Утилиты **jobs, fg, bg** предназначены для перевода выполнения процессов в фоновый режим или режим переднего плана.

Синтаксис утилит управления процессами приведён в приложении.

3 Выполнение работы

3.1 Задание

Познакомиться с возможностями утилит Alt Linux для решения задач мониторинга и управления процессами операционной системы.

Выполнить команды управления процессами и объяснить полученные результаты.

3.2 Порядок выполнения работы

1. Войти в систему под учётной записью **user2**
2. Запустить программу виртуализации **Oracle VM VirtualBox**
3. Запустить виртуальную машину **Alt10-XX**
4. Войти в систему под учётной записью **admin_kaf**.

3.2.1 Выполнить мониторинг и управление процессами с помощью монитора процессов *htop*

1. Запустить программу **VirtualBox**
2. Запустить программу «Терминал»
3. Определить **PID** программы **VirtualBox**
4. Запустить монитор **htop** для отображения информации о процессе **VirtualBox**

5. Проанализировать основные параметры процесса **VirtualBox** и всех дочерних процессов (общий объём виртуальной памяти, объём резидентной памяти, объём разделяемой памяти (общих страниц))
6. Определить приоритеты **планирования** и приоритеты **nice**, присвоенные процессу **VirtualBox** и дочерним процессам при запуске программы
7. Повысить приоритет **nice** процесса **VirtualBox** и всех дочерних процессов до максимального
8. Завершить выполнение монитора

3.2.2 *Выполнить мониторинг и управление процессами с помощью графической утилиты «Системный монитор»*

1. Запустить программу «**blender**»
2. Запустить программу «**Системный монитор**»
3. Перейти на вкладку «**Система**» и проанализировать полученную информацию
4. Перейти на вкладку «**Процессы**»
5. Выделить процесс «**blender**»
6. С помощью контекстного меню просмотреть информацию о процессе и изменить приоритет процесса до **Высокий**
7. Используя вкладку **Ресурсы** отобразить и проанализировать информацию о системных ресурсах
8. Используя вкладку **Файловые системы** отобразить информацию об устройствах и файловых системах
9. Объяснить полученные результаты

3.2.3 *Выполнить мониторинг и управление процессами с помощью утилит командной строки*

1. Открыть окна интерпретатора команд
2. Получить общую информацию о системе
 - вывести информацию о текущем интерпретаторе команд
 - вывести информацию о текущем пользователе
 - вывести информацию о текущем каталоге

- вывести информацию об оперативной памяти и области подкачки
- вывести информацию о файловой системе

3. Получить информации о процессах

- получить идентификатор текущего процесса(**PID**)
- получить идентификатор родительского процесса(**PPID**)
- получить идентификатор процесса по его имени(**init**)
- получить информацию о выполняющихся процессах с помощью команды **ps**

4. Выполнить команды управления процессами

- получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе
- определить текущее значение приоритета **nice** по умолчанию
- определить **PID** запущенного командного процессора
- установить приоритет **nice** командного процессора равным **5**

4 Контрольные вопросы

1. Как создаются задачи в ОС Alt Linux?
2. Назовите состояния задачи в ОС Alt Linux.
3. Назовите классы потоков ОС Alt Linux
4. Как используется приоритет планирования при запуске задачи
5. Как можно изменить приоритет для выполняющейся задачи?

5 ЛИТЕРАТУРА

1. Робачевский А.М. Операционная система UNIX.-СПб.: БХВ-Петербург, 2001. – 528 с.:ил.
2. Негус К. Ubuntu и Debian Linux для продвинутых. 2-е изд. – СПб.: Питер,2014. -384 с.: ил.

6 Приложение.

6.1 Монитор процессов htop

6.1.1 Запуск монитора

\$ sudo htop <параметры>

После запуска открывается окно программы, разделённое на две части:

- **общая информация о системе**
- **подробная информация о процессах**

6.1.2 Область с общей информацией о системе

CPU - нагрузка на каждое ядро центрального процессора (цифры от 1 до 12).

Mem - общее количество оперативной памяти и используемая память.

Swp – статистика файла подкачки (если он есть)

Task - статистика по процессам

Load average - средняя загрузка центрального процессора

Uptime - время работы операционной системы с момента последней загрузки

6.1.3 Область с подробной информацией о процессах.

Значение столбцов:

PID - идентификатор процесса.

USER - имя пользователя-владельца процесса или ID если имя не может быть определено.

PRI - приоритет планирования - внутренний приоритет ядра для процесса. Значение по умолчанию равно **20**. Равен значению приоритета процесса **nice** плюс двадцать.

Приоритет планирования меняется от **0** до **39**.

Чем меньше приоритет планирования, тем больше времени отводится процессу (отличается для процессов, имеющих приоритет выполнения real-time).

NI - приоритет **nice** процесса. Значение от **19** (низкий приоритет) до **-20** (высокий приоритет). Может быть изменено клавишами **F7** и **F8**.

VIRT – общий объём виртуальной памяти процесса (**M_SIZE**). Включает в себя: область кода (**CODE**), данные (**DATA**), разделяемые библиотеки (**SHARED**) и страницы, перемещенные в swar-область памяти (**CODE** - объем памяти, содержащий исполняемый код процесса. **DATA**- объем памяти, занятой данными, используемыми процессом в ходе выполнения. **SWAP** - объем памяти, используемой процессом, но перемещенной в swar-область.)

RES - объём резидентной (не перемещаемой в swar) памяти (text + data + stack) процесса (размер используемой физической памяти процесса, **M_RESIDENT**).

SHR – объём общих страниц процесса (**M_SHARE**) т.е. памяти, которая может быть использована другими приложениями.

S (**STATE**) - состояние процесса:

S - для спящих (в простое);

R - для запущенных (в состоянии выполнения);

D – для ожидающих;

Z для зомби;

CPU% - процент процессорного времени, которое процесс использует в данный момент.

MEM% - процент памяти, используемой процессом в данный момент (в зависимости от размера резидентной памяти процесса (**M_RESIDENT**)).

TIME+ - время, измеренное в часах, указывает на то, сколько процесс провёл в пользовательском и системном режимах.

Command – полный путь к программе процесса (имя программы и аргументы).

6.1.4 Назначение функциональных клавиш

F1 - справка;

F2 - настройки;

F3 - поиск процесса;

F4 - сортировка списка процессов;

F5 - вывод процессов в виде дерева;

F6 - выбор параметра для сортировки;

F7(Nice -) - увеличить приоритет **nice** выделенного процесса;

F8(Nice +) - уменьшить приоритет **nice** выделенного процесса;

F9 – убить процесс;

F10 – выйти из программы.

6.1.5 Клавиши для перемещения по окну программы

Клавиши **↑ ↓** - для прокрутки списка процессов.

Клавиши **PgUp PgDn** - для прокрутки окна.

Кнопка **Home** - перемещение к началу списка.

Кнопка **End** – перемещение к концу списка.

Ctrl-a - прокрутка к началу строки.

Ctrl-e - прокрутка к концу строки.

6.1.6 Поиск и фильтрация процессов

Поиск отличается от фильтрации тем, что найденные процессы показываются наравне с остальными, и между найденными процессами можно переключаться кнопкой **F3**.

При фильтрации на экран будут выводиться только процессы, соответствующие введённой строке.

Для перехода к поиску по процессам надо нажать **F3**. Для переключения между найденными процессами нажать **F3**.

Для фильтрации процессов нажать **F4** и ввести имя процесса.

Для очистки фильтра вновь нажать **F4** и затем Esc.

6.1.7 Изменение приоритета процесса

Для увеличения приоритета процесса (вычитание из величины **nice**) необходимо нажать **F7**. Эту операцию может выполнять только суперпользователь (пользователь **root** или если **htop** запущена с использованием утилиты sudo).

Для уменьшения приоритета процесса (прибавления к величине nice) надо нажать **F8**.

6.1.8 Выбор нескольких процессов

Для выбора процессов необходимо использовать **Пробел**. После этого введённые команды, такие как **kill** или **изменение приоритета**, могут применяться к группе выделенных процессов вместо подсвеченного в данный момент. Для снятия выделения со всех процессов нажать **U (Shift+u)**.

6.1.9 Закрытие процесса

Для закрытия процесса надо выбрать один или несколько процессов и нажать **F9** или. Выбранному процессу будет отправлен сигнал завершения. Если не отмечен ни один процесс, то будет закрыт тот, на котором находится в данный момент курсор.

6.1.10 *Отображение только процессов определённого пользователя*

```
$ sudo htop -u <имя пользователя>
```

6.1.11 *Отображение только процессов с определёнными номерами*

```
$ sudo htop -p <PID,PID....>
```

6.2 Утилита «Системный монитор»

Для запуска Системного монитора следует перейти

Меню > Приложения > Системные > Системный монитор МАТЕ.

Вся информация в окне программы распределена по четырем вкладкам:

6.2.1 Вкладка «Система»

Выводится основная информация об установленной операционной системе и оборудовании:

- имя компьютера
- версия ОС
- версия ядра Linux
- объём памяти

- информация о процессоре
- информация о доступном дисковом пространстве

6.2.2 Вкладка «Процессы»

Вкладка позволяет просматривать и управлять запущенными процессами. Каждый процесс можно **приостановить, остановить, изменить приоритет** и выполнить некоторые другие действия;

При щелчке правой кнопкой мыши по любому запущенному процессу, открывается контекстное меню, с помощью которого можно завершить «зависшее» приложение, остановить, перезапустить и изменить его приоритет, что позволяет регулировать требования процесса к системным ресурсам.

Для изменения приоритета процесса необходимо:

1. Выбрать вкладку **Процессы**, чтобы отобразить список процессов;
2. Выбрать процесс, приоритет которого следует изменить.
3. В контекстном меню процесса выбрать пункт **Изменить приоритет**.
4. Если выбрать пункт **Вручную**, откроется диалоговое **окно Изменить приоритет процесса**.
5. Можно использовать ползунок, чтобы установить уровень приоритета. Приоритет процесса задается уровнем **nice**. Меньшее значение **nice** соответствует более высокому приоритету.
6. Нажать кнопку **Изменить приоритет**.

Для установки более высокого приоритета, чем тот, который уже установлен у процесса, потребуется ввести пароль пользователя, находящегося в группе **wheel**.

6.2.3 Вкладка «Ресурсы»

В данном окне в реальном времени выводится информация о ресурсах (в виде графиков) – использование процессора (CPU), использование оперативной памяти (RAM) и файла подкачки (SWAP), использование сети;

6.2.4 Вкладка «Файловые системы»

С помощью данной вкладки можно просматривать информацию о всех файловых системах ОС.

Отображается информация о типе файловой системы, общий объём и объём доступного пространства.

6.3 Команды интерпретатора BASH мониторинга и управления процессами

6.3.1 Команды получения общей информации о системе

\$ echo \$SHELL - информация о текущем интерпретаторе

\$ whoami - информация о текущем пользователе

\$ pwd - информации о текущем каталоге

\$ free - информация об оперативной памяти и файле подкачки

\$ df - информация о afqkjds[cbcntvf[

6.3.2 Команды получения информации о процессах

\$ echo \$\$ - информация об идентификаторе текущего процесса(PID)

\$ echo \$PPID - информация об идентификаторе родительского процесса(PPID)

\$ pidof <имя процесса> - информация об идентификатор процесса по его имени

\$ pgrep -l <имя процесса> - информация об идентификатор процесса по его имени

\$ ps <параметр> - информация о выполняемых процессах

Параметр	Значение
Без параметра	информации о выполняемых процессах текущего пользователя в текущем интерпретаторе команд
-a	отобразить все процессы, связанных с терминалом (отображаются процессы всех пользователей)
-e	отобразить все процессы
-t <список терминалов>	отобразить процессы, связанные с терминалами
-u <имена пользователей>	отобразить процессы, связанные с данными пользователями
-g <имена групп>	отобразить процессы, связанные с данными группами пользователей

-x	отобразить все процессы, не связанные с терминалом
-----------	--

6.3.3 Команды управления процессами

\$ nice - определение текущего значения приоритета **nice** по умолчанию

\$ sudo renice -n <значение nice> -p <PID> - изменение приоритета выполняемого процесса

\$ sudo kill <-номер сигнала> <PID> – принудительное завершение процесса