

Министерство образования и науки Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ЭКОНОМИКИ И УПРАВЛЕНИЯ «НИНХ»
Кафедра информационных технологий

**МЕТОДИЧЕСКОЕ РУКОВОДСТВО
ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ
СТУДЕНТОВ ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ**

Учебная дисциплина
Системное программирование

по направлению подготовки 09.03.02 «Информационные системы и технологии», без профиля

Квалификация выпускника: «бакалавр»

Форма обучения: очная

Новосибирск 2016

ОГЛАВЛЕНИЕ

РАЗДЕЛ 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ.....	4
1.1. Организация самостоятельной работы студентов по подготовке к лабораторным занятиям.....	4
1.2. Содержание лабораторных занятий.....	4
Тема 1.1. Основные понятия и определения ОС Linux	4
Тема 1.2. Инструментальные средства ОС Linux	5
Тема 1.3. Основные конструкции языка С	7
Тема 2.1. Базовые операции ввода-вывода.....	8
Тема 2.2. Дополнительные операции ввода-вывода.....	9
Тема 2.3. Командный интерпретатор shell и пользовательские интерфейсы.....	10
Тема 3.1. Системные вызовы работы с процессами.....	11
Тема 3.2. Системные вызовы работы с потоками.....	12
Тема 3.3. Системное администрирование	13
Тема 4.1. Именованные и неименованные каналы.....	14
Тема 4.2. Системные вызовы для работы с очередями.....	15
Тема 4.3. Семафоры и системные вызовы для работы с общей памятью.....	17
Тема 5.1. Сокеты.....	18
Тема 5.2. Сигналы.....	19
1.3. Список библиографических источников для подготовки к лабораторным занятиям по разделам учебной дисциплины.....	20
РАЗДЕЛ 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЗАПЛАНИРОВАННЫХ ВИДОВ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.....	22
2.1. Методические указания по выполнению расчетно-графической работы.....	22
2.1.1. Задания для выполнения расчетно-графической работы.....	22
2.1.2. Порядок выбора варианта расчетно-графической работы.....	22
2.1.3. Указания на сроки выполнения и защиты расчетно-графической работы.....	22
2.1.4. Требования к структуре и содержанию расчетно-графических работ	22
2.1.5. Критерии оценки расчетно-графической работы.....	23
2.1.6. Требования к форме представления результатов, оформлению титульного листа и текста расчетно-графической работы.....	23
РАЗДЕЛ 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	24
3.1. Список вопросов для подготовки к зачету.....	25
3.2. Общие положения проведения зачета.....	25
Приложение 1	27
Приложение 2	31

РАЗДЕЛ 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ

1.1. Организация самостоятельной работы студентов по подготовке к лабораторным занятиям

Студенту рекомендуется следующая схема подготовки к лабораторному занятию по учебной дисциплине «Системное программирование»:

1. Проработать конспект лекций.
2. При необходимости обратиться к источникам основной и дополнительной литературы, рекомендованной по каждому из трех разделов учебной дисциплины.
3. Подготовить ответы на вопросы, входящие в структуру содержания лабораторного занятия по каждой теме соответствующего раздела учебной дисциплины.
4. Ответить на вопросы тестовых заданий по каждой конкретной теме соответствующего раздела учебной дисциплины, выбрав один или несколько вариантов ответа верных, по вашему мнению.
5. При затруднениях сформулировать вопросы к преподавателю.

Формой текущего контроля самостоятельного изучения студентом отдельных тем является тестирование.

1.2. Содержание лабораторных занятий

Лабораторные занятия по дисциплине «Системное программирование» проводятся в соответствии с учебно-тематическим планом и планом лабораторных занятий, отраженным в Рабочей программе, утвержденной на заседании кафедры прикладных информационных технологий 7 июня 2011 г., протокол № 11.

Раздел 1. Введение в ОС Linux

Тема 1.1. Основные понятия и определения ОС Linux

1. Изучение архитектуры ОС Linux. Структура ОС Linux
2. Файловая система ОС Linux.

Вопросы и задания для самостоятельной работы:

1. Подготовить ответы на контрольные вопросы по теме:
 - 1.1. Каковы особенности UNIX?
 - 1.2. Что такое файловая система?
 - 1.4. Как создать каталог?
 - 1.5. Как организуются права доступа к файлам и каталогам?
 - 1.6. Изучите физическую структуру файловой системы.

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Какие из следующих шаблонных символов не поддерживаются Linux:

- *
- [...]
- &
- #
- >

2.2. Какая из следующих команд используется в Linux для активизации приостановленного процесса:

- Ctrl+Z
- jobs
- fg
- Bg
- Kill
- ps

2.3. Какой из следующих каталогов Linux содержит разделяемые системные библиотеки

- /bin
- /usr/bin
- /usr/spool
- /usr/lib

2.4. Что содержится в каталоге /dev системы Linux:

- Основные исполняемые программы Linux
- Каталоги, предназначенные для пользователей
- Конфигурационные файлы
- Файлы устройств

2.5. Что выполняет следующая команда `chmod ug-r Research`:

- Изменяет режим файла
- Изменяет каталог
- Изменяет владельца файла
- Изменяет права доступа к файлу

Тема 1.2. Инструментальные средства ОС Linux

1. Фильтры. Команда `rg`.
2. Монтирование файловых систем.

Вопросы и задания для самостоятельной работы:

1. Подготовить ответы на контрольные вопросы по теме:

- 1.1. Изучите команда `rg` и ее ключи.
- 1.2. Изучите команда `sort` и ее ключи.

- 1.3. Какие функции выполняют команды `grep`, `fgrep`, `egrep`?
- 1.4. Что делает команда `make`?
- 1.5. Как происходит монтирование файловых систем

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Какие четыре из следующих переменных существуют в Linux?

- `SCRIPT`
- `USERNAME`
- `HISTSIZE`
- `SHELL`
- `PS1`

2.2. Какая команда используется для уничтожения задачи с номером 27

- `kill-27`
- `kill&27`
- `kill%27`
- `kill-9 27`
- `ps 27`
- `bg 27`

2.3. Какой символ используется для переназначения стандартного вывода программы:

- `>`
- `<`
- `>>`
- `~`
- `&`
- `%`

2.4. Какая команда используется для получения списка переменных окружения оболочки `bash`:

- `ps`
- `jobs`
- `more`
- `set`
- `less`
- `cat`

2.5. Что означает запись `drwxrwx-r-x` в списке выводимом командой `ls -l`:

- соответствующий файл является каталогом
- устройство ввода-вывода
- права доступа к файлу
- соответствующий файл является каталогом и указываются права доступа к нему для пользователя, группы и всех остальных

Тема 1.3. Основные конструкции языка C

1. Основные типы данных
2. Написание программ для задач сортировки массива

Вопросы и задания для самостоятельной работы:

1. Подготовить ответы на контрольные вопросы по теме:
 - 1.1. Каковы элементарные типы данных?
 - 1.2. Рассмотрите основные принципы работы с указателями?
 - 1.3. Укажите механизм работы рекурсивных подпрограмм.
 - 1.4. Укажите способы передачи параметров в подпрограмму в различных языках программирования.
 - 1.5. Как происходит распределение динамической памяти?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

- 2.1. Что такое цикл?
 - Имя переменной
 - Оператор присваивания
 - Оператор предназначенный для многократного использования определенных инструкций
 - Имя подпрограммы
- 2.2. Оператор условия иногда называют, оператором:
 - Присваивания
 - Ветвления
 - Просто оператором
 - Высказывания
- 2.3. Зачем нужен тип данных:
 - Для обеспечения целостности
 - Для создания переменных с неизвестным типом
 - Для указания переменной типа ее содержимого
 - Для быстроты вычислений
- 2.4. Что такое массив:
 - Именованный набор переменных имеющих различные типы данных, и располагающихся в одной памяти
 - Именованный набор переменных и функций, которые располагаются в одной области памяти
 - Именованный набор переменных имеющих один тип данных, и располагающихся в одной области памяти
 - Неименованный набор переменных
- 2.5. Как написать следующее выражение "Первому элементу массива array присвоено значение пять":
 - `int array[0]="пять";`
 - `int array [0] = 5;`

- `int array [0] = "пять";`
- `int array = 5;`

Раздел 2. Операции ввода-вывода

Тема 2.1. Базовые операции ввода-вывода

1. Стандартные дескрипторы. Программирование с помощью системных вызовов `open` и `creat`.
2. Системный вызов `close`. Системный вызов `lseek`. Системные вызовы `pread` и `pwrite`.

Вопросы и задания для самостоятельной работы:

1. Подготовить ответы на контрольные вопросы по теме:

- 1.1. Что такое стандартные дескрипторы?
- 1.2. Изучите основные системные вызовы ввода-вывода.
- 1.3. В чем суть системного вызова `lseek`?
- 1.4. Что такое текущая позиция в файле?
- 1.5. Каковы функции системного вызова `unlink`.

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Дескриптор с номером 0 соответствует:

- Устройству стандартного ввода
- Устройству стандартного вывода
- Устройству стандартного вывода сообщений об ошибках
- Устройству печати

2.2. Дескриптор с номером 1 соответствует:

- Устройству стандартного ввода
- Устройству стандартного вывода
- Устройству стандартного вывода сообщений об ошибках
- Устройству печати

2.3. Дескриптор с номером 2 соответствует:

- Устройству стандартного ввода
- Устройству стандартного вывода
- Устройству стандартного вывода сообщений об ошибках
- Устройству печати

2.4. Создание нового файла осуществляется системным вызовом:

- `Open`
- `Write`
- `Creat`
- `Unlink`

2.5. Чтение из файлового дескриптора выполняет системный вызов:

- Open
- Write
- Creat
- Unlink
- Read

Тема 2.2. Дополнительные операции ввода-вывода

1. Каталоги. Системные вызовы `chdir` и `fchdir`
2. Системные вызовы `chown`, `fchown` и `lchown`. Системный вызов `utime`. Системный вызов `mknod`.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое жесткие и символические ссылки?
- 1.2. Как происходит отображение метаданных?
- 1.3. В чем суть системного вызова `mknod`?
- 1.4. Изучите системные вызовы `chown`, `fchown` и `lchown`.
- 1.5. Опишите работу с каталогами.

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Какой системный вызов создает жесткую ссылку:

- Open
- Write
- Link
- Unlink

2.2. Какой системный вызов переименовывает файлы и каталоги:

- Rename
- Write
- Link
- Unlink

2.3. Возвращает полное имя текущего каталога системный вызов:

- Rename
- Getcwd
- Link
- Unlink

2.4. Возвращает сведения о файле по его имени системный вызов:

- Rename
- Getcwd
- Stat

- Unlink

2.5. Возвращает сведения о файле по дескриптору системный вызов:

- Rename
- Getcwd
- Stat
- Fstat

Тема 2.3. Командный интерпретатор shell и пользовательские интерфейсы

1. Команды shell. Функциональные возможности shell. Командная среда. Операторы командного языка. Командные файлы. Программирование в shell.
2. Графические пользовательские интерфейсы. X Window System. X - серверы.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое командная среда?
- 1.2. Как происходит программирование в shell.
- 1.3. Как происходит перенаправление фоновых процессов?
- 1.4. Зачем нужны графические пользовательские интерфейсы?
- 1.5. Опишите функциональные возможности shell.

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Можно ли включить в программу на языке C системный вызов sort:

- Да
- Нет
- Только вместе с системным вызовом exec
- Только вместе с системным вызовом fork

2.2. Можно ли включить в программу на языке C системный вызов ls:

- Да
- Нет
- Только вместе с системным вызовом exec
- Только вместе с системным вызовом wait

2.3. Можно ли включить в программу на языке C системный вызов dup2:

- Да
- Нет
- Только вместе с системным вызовом exec
- Только вместе с системным вызовом wait

2.4. Новый процесс создается системным вызовом:

- Rename
- Getcwd
- Stat
- Fork

2.5. Завершение процесса осуществляется системным вызовом:

- Rename
- Exit
- Stat
- Fstat

Раздел 3. Системное программирование

Тема 3.1. Системные вызовы работы с процессами

1. Системный вызов `exec`. Системный вызов `fork`. Завершение процесса и системные вызовы `exit`. Системные вызовы `wait`, `waitpid` и `waited`.
2. Получение и изменение идентификаторов пользователя и группы. Получение и изменение приоритета. Ограничения для процессов.

Вопросы и задания для самостоятельной работы:

- 1.1. Как создаются новые процессы?
- 1.2. Как завершаются процессы?
- 1.3. Как изменить приоритет?
- 1.4. Изучите основные системные вызовы работы с процессами

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите системный вызов, который ожидает завершения дочернего процесса:

- Rename
- Getcwd
- Stat
- Fork
- Wait

2.2. Укажите системный вызов, который возвращает реальный идентификатор пользователя:

- Rename
- Getcwd
- Getuid
- Fork

- Wait

2.3. Укажите системный вызов, который возвращает действующий идентификатор пользователя:

- Rename
- Getcwd
- Getuid
- Fork
- Geteuid

2.4. Укажите системный вызов, который возвращает действующий идентификатор группы:

- Rename
- Getcwd
- Getuid
- Getegid
- Geteuid

2.5. Укажите системный вызов, который возвращает реальный идентификатор группы:

- Getgid
- Getcwd
- Getuid
- Getegid
- Geteuid

Тема 3.2. Системные вызовы работы с потоками

1. Создание потока. Ожидание завершения потока.
2. Синхронизация потоков. Переменные состояния.
3. Проблема блокирования. Решения с использованием потоков и процессов. Диспетчер событий

Вопросы и задания для самостоятельной работы:

- 1.1. Как создаются потоки?
- 1.2. Как происходит синхронизация?
- 1.3. Как разрешается система блокировки?
- 1.4. Что такое события?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Создание нового потока осуществляется системным вызовом:

- Getgid
- Getcwd
- Pthread_create

- Getegid
- Geteuid

2.2. Укажите системный вызов для работы с мьютексами:

- Getgid
- Getcwd
- Pthread_create
- Pthread_mutex_lock
- Geteuid

2.3. Укажите системный вызов, который сигнализирует о наступлении состояния:

- Getgid
- Pthread_cond_signal
- Pthread_create
- Pthread_mutex_lock
- Geteuid

2.4. Укажите системный вызов, который ожидает наступления состояния:

- Pthread_cond_wait
- Pthread_cond_signal
- Pthread_create
- Pthread_mutex_lock
- Geteuid

2.5. Укажите системный вызов, который устанавливает обработчик принудительного завершения:

- Pthread_cond_wait
- Pthread_cond_signal
- Pthread_create
- Pthread_mutex_lock
- Pthread_cleanup_push

Тема 3.3. Системное администрирование

1. Поддержка файловой системы. Целостность файловой системы.
2. Создание резервных копий файловых систем. Поддержка пользовательских учетных записей
3. Установка драйвера устройства. Терминальные файлы. Сетевой интерфейс. Учет системных ресурсов. Конфигурирование ядра. Проблема безопасности.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое целостность файловой системы?
- 1.2. Что такое драйвер устройства?
- 1.3. Как учитываются системные ресурсы?
- 1.4. Как происходит конфигурирование ядра?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите файл, который содержит в себе информацию о паролях пользователей:

- Passwd
- Shadow
- Group
- Ust

2.2. Укажите файл, который содержит в себе информацию обо всех пользователях:

- Passwd
- Shadow
- Group
- Ust

2.3. Укажите файл, который описывает группы пользователей:

- Passwd
- Shadow
- Group
- Ust

2.4. Добавление нового пользователя осуществляет команда:

- Pthread_cond_wait
- Pthread_cond_signal
- Pthread_create
- Pthread_mutex_lock
- User_add

2.5. Удаление пользователя осуществляет команда:

- Pthread_cond_wait
- Pthread_cond_signal
- User_del
- Pthread_mutex_lock
- User_add

Раздел 4. Механизмы межпроцессного взаимодействия

Тема 4.1. Именованные и неименованные каналы

1. Каналы. Системные вызовы dup и dup2. Командная оболочка.
2. Двухнаправленное взаимодействие с использованием однонаправленных каналов.
3. Именованные каналы (FIFO). Создание именованных каналов. Достоинства и недостатки именованных каналов.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое именованные каналы?
- 1.2. Как происходит взаимодействие процессов через каналы?
- 1.3. Каковы функции системных вызовов `dup` и `dup2`?
- 1.4. Каковы достоинства и недостатки именованных каналов?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите системный вызов для работы с каналами:

- `Getgid`
- `Getcwd`
- `Pthread_create`
- `Pipe`
- `Geteuid`

2.2. Результат конвейера `who | sort | more` выдает:

- Список пользователей
- Отсортированный список пользователей
- Отсортированный список пользователей, вошедших в систему
- Все процессы

2.3. Укажите системный вызов, который дублирует дескриптор файла:

- `Rename`
- `Getcwd`
- `Getuid`
- `Dup`
- `Geteuid`

2.4. Результат конвейера `who | wc` выдает:

- Все процессы
- Число пользователей
- Число пользователей, вошедших в систему
- Число выполняемых процессов

2.5. Укажите системный вызов, который создает именованный канал:

- `Getgid`
- `Mkfifo`
- `Getuid`
- `Getegid`
- `Geteuid`

Тема 4.2. Системные вызовы для работы с очередями

1. Системные вызовы для работы с очередями сообщений POSIX: `mq_open`; `mq_close`
2. Системные вызовы для работы с очередями сообщений POSIX: `mq_unlink`; `mq_send`
3. Системные вызовы для работы с очередями сообщений POSIX: `mq_receive`; `mq_notify`.

Вопросы и задания для самостоятельной работы:

1.1. Что такое очередь?

1.2. Изучите основные системные вызовы для работы с очередями.

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите системный вызов, который возвращает идентификатор очереди сообщений:

- `Getgid`
- `Getcwd`
- `Msgget`
- `Pipe`
- `Geteuid`

2.2. Укажите системный вызов, который управляет очередью сообщений:

- `Getgid`
- `Msgctl`
- `Msgget`
- `Pipe`
- `Geteuid`

2.3. Укажите системный вызов, который помещает сообщение в очередь:

- `Rename`
- `Getcwd`
- `Getuid`
- `Dup`
- `Msgsnd`

2.4. Укажите системный вызов, который извлекает сообщение в очередь:

- `Rename`
- `Getcwd`
- `Msgrcv`
- `Dup`
- `Msgsnd`

2.5. Укажите системный вызов, который открывает очередь сообщений:

- `Mq_open`

- Mkfifo
- Getuid
- Getegid

Тема 4.3. Семафоры и системные вызовы для работы с общей памятью

1. Семафоры. Основные принципы использования семафоров. Системные вызовы для работы с семафорами: `sem_init`; `sem_destroy`
2. Именованные семафоры. Неименованные семафоры. Блокировка к доступа к файлу с помощью семафоров.
3. Системные вызовы для работы с общей памятью POSIX: `shm_open`; `ftruncate`; `mmap`.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое семафоры?
- 1.2. Изучите основные системные вызовы для работы с семафорами.
- 1.3. Как работают именованные семафоры?
- 1.4. Как работают неименованные семафоры?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите системный вызов, который возвращает идентификатор набора семафоров:

- Getgid
- Getcwd
- Msgget
- Pipe
- Semget

2.2. Укажите системный вызов, который управляет набором семафоров:

- Getgid
- Msgctl
- Msgget
- Semctl
- Geteuid

2.3. Укажите функцию, которая открывает семафор:

- Rename
- Getcwd
- Getuid
- Dup
- SimpleSemOpen

2.4. Укажите функцию, которая закрывает семафор:

- SimpleSemClose
- Getcwd
- Getuid
- Dup
- SimpleSemOpen

2.5. Укажите функцию, которая удаляет семафор:

- SimpleSemClose
- Getcwd
- SimpleSemRemove
- Dup
- SimpleSemOpen

Раздел 5. Сетевое взаимодействие

Тема 5.1. Сокеты

1. Организация обмена информацией между компьютерами, объединенными в сеть. Понятие сокета. Принцип действия сокетов
2. Основные системные вызовы для работы с сокетами, образующими логические соединения. Обслуживание нескольких клиентов. Адресация сокетов.
3. Сокеты, не требующие создания соединения. Дейтаграммы.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое сокеты?
- 1.2. Как происходит адресация сокетов?
- 1.3. Что такое дейтаграммы?
- 1.4. Как происходит обслуживание нескольких клиентов?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите системный вызов, который присваивает адрес сокету:

- Getgid
- Getcwd
- Pthread_create
- Bind
- Geteuid

2.2. Укажите системный вызов, который создает объект для организации сетевых взаимодействий:

- Getgid
- Getcwd

- Socket
- Bind
- Geteuid

2.3. Укажите системный вызов, который подготавливает сокет к приему запросов на соединение:

- Listen
- Getcwd
- Getuid
- Dup
- Geteuid

2.4. Укажите системный вызов, который принимает новое соединение и создает новый сокет:

- Getgid
- Mkdir
- Accept
- Getegid
- Geteuid

2.5. Укажите функцию, которая возвращает информацию об адресе сокета:

- Getgid
- Mkdir
- Getuid
- Getegid
- Getaddrinfo

Тема 5.2. Сигналы

1. Жизненный цикл сигналов. Типы сигналов. Прерывание системных вызовов.
2. Системные вызовы для работы с сигналами. Обработчики сигналов. Генерация сигналов. Ожидание сигнала. Сигналы реального времени и их обработчики.
3. Часы и таймеры. Основные системные вызовы для работы с таймерами.

Вопросы и задания для самостоятельной работы:

- 1.1. Что такое сигнал?
- 1.2. Изучите основные системные вызовы для работы с сигналами.
- 1.3. Как происходит генерация сигналов?
- 1.4. Как обрабатываются сигналы?

2. Тестовые задания для самостоятельного контроля уровня подготовки студентами вопросов темы:

2.1. Укажите системный вызов, который устанавливает реакцию приложения на сигнал:

- Getgid
- Getcwd
- Sigaction
- Pipe
- Geteuid

2.2. Укажите системный вызов, который посылает сигнал процессу:

- Getgid
- Msgctl
- Kill
- Pipe
- Geteuid

2.3. Укажите системный вызов, который посылает сигнал группе процессов:

- Getgid
- Killpg
- Kill
- Pipe
- Geteuid

2.4. Укажите системный вызов, который посылает сигнал потоку:

- Getgid
- Pthread_kill
- Kill
- Pipe
- Geteuid

2.5. Укажите системный вызов, который ожидает доставки сигнала:

- Sigwait
- Mkfifo
- Getuid
- Getegid

1.3. Список библиографических источников для подготовки к лабораторным занятиям по разделам учебной дисциплины

1.3.1. Библиографические обязательные источники для подготовки к лабораторным занятиям

б) учебные пособия:

1. Федотова, Е. Л. Информационные технологии в профессиональной деятельности : учеб. пособие / Е. Л. Федотова.- М.: ФОРУМ, 2008 .- 366 с. (МОРФ)

2. Попов, В. Б. Основы информационных и телекоммуникационных технологий: мультимедиа: учеб. пособие / В. Б. Попов.- М.: Финансы и статистика, 2007 .- 333 с. (УМО)

3. Партыка, Т. Л. Операционные системы, среды и оболочки: учеб. пособие / Т. Л. Партыка, И. И. Попов .- 2-е изд., испр. и доп .- М.: ФОРУМ, 2007 .- 527 с. (МОРФ)

4. Джонсон, Майкл К. Разработка приложений в среде Linux = Linux application development : [пер. с англ.] / Майкл К. Джонсон, Эрик В. Троан .- 2-е изд .- М. : Вильямс, 2007 .- 543 с .

5. Лупин, С. А. Технологии параллельного программирования: учеб. пособие для вузов / С. А. Лупин, М. А. Посыпкин .- М.: ФОРУМ, 2008 .- 205 с .

в) методическая литература:

1. Методические указания по организации самостоятельной работы студентов очной формы обучения по направлению подготовки 230400.62 «Информационные системы и технологии», без профиля.

Информационные источники: www.intuit.ru

Программные продукты: операционная система Linux; транслятор с языка программирования С.

1.3.2. Библиографические дополнительные источники для подготовки к лабораторным занятиям

б) учебные пособия:

1. Истомин, Е. П. Высокоуровневые методы информатики и программирования : учебник / Е. П. Истомин, В. В. Новиков, М. В. Новикова ; Рос. гос. гидрометеорол. ун-т .- СПб. : Андреев. изд. дом, 2006 .- 228 с .(УМО)

2. Губенков А.А. Информационная безопасность вычислительных сетей: учеб. пособие / А.А. Губенков. – Саратов: СГТУ, 2009. – 88 с.

3. Подбельский, В. В. Стандартный Си++ : учеб. пособие для высш. учеб. / В. В.Подбельский .- М. : Финансы и статистика, 2008 .- 687 с.(УМО)

4. Молчанов, А. Ю. Системное программное обеспечение: лабораторный практикум / А. Ю. Молчанов .- СПб. : Питер, 2005 .- 283 с.

г) научная литература:

1. Собель, М.Г. Linux. Администрирование и системное программирование / М.Г. Собель. - СПб. : Питер, 2011 .- 880 с.

2. Лав, Роберт Linux. Системное программирование / Роберт Лав. - СПб. : Питер, 2008 .- 416 с.

Информационные источники: поисковые порталы Yandex, Googl, Rambler и др.

Программные продукты: транслятор с языка программирования С++.

РАЗДЕЛ 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЗАПЛАНИРОВАННЫХ ВИДОВ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Согласно Рабочему учебному плану подготовки студентов очной формы обучения по направлению подготовки 230400.62 «Информационные системы и технологии», без профиля организация самостоятельной работы студентов заключается в подготовке к лабораторным занятиям, выполнению двух расчетно-графических работ, охватывающих все темы.

2.1. Методические указания по выполнению расчетно-графической работы «Системное программирование»

2.1.1. Задания для выполнения расчетно-графической работы

Целью расчетно-графической работы является изложение теоретических аспектов теории системного программирования и практических результатов применения их в разработке средств проектирования и реализации информационных технологий.

В процессе выполнения расчетно-графической работы студенты должны приобрести практические навыки с операционной системой Linux.

Каждая из расчетно-графических работ по дисциплине «Системное программирование» состоит из одного задания, описывающего следующие разделы курса: введение в ОС Linux; операции ввода-вывода; системное программирование; механизмы межпроцессного взаимодействия; сетевое взаимодействие.

2.1.2. Порядок выбора варианта расчетно-графической работы

Номер варианта расчетно-графической работы определяется преподавателем.

Варианты расчетно-графических работ на текущий учебный год представлены в Приложении 1.

2.1.3. Указания на сроки выполнения и защиты расчетно-графической работы

Расчетно-графическая работа сдается на кафедру в печатном виде, а затем передается преподавателю на проверку. В случае отметки «к защите» работа защищается студентом в назначенное преподавателем время. В случае отметки «на доработку» студент устраняет недостатки и повторно сдает исправленную работу на кафедру. После защиты расчетно-графической работы студент допускается к сдаче зачета по дисциплине «Системное программирование».

2.1.4. Требования к структуре и содержанию расчетно-графических работ

Введение. Описывается цель расчетно-графической работы и дается краткое изложение теоретических основ решаемых заданий.

Основная часть. Выполняются задания расчетно-графической работы.

Библиографический список. В библиографический список включаются названия учебников, пособий, журналов, электронные документы и т.д., которые

использовались при выполнении расчетно-графической работы.

Библиографический список оформляется согласно ГОСТ 7.1 – 2003. «Библиографическая запись. Библиографическое описание».

2.1.5. Критерии оценки расчетно-графической работы

При защите расчетно-графической работы студент должен уметь объяснить методы и подходы, используемые при решении практических заданий и ответить на дополнительные вопросы преподавателя, касающиеся рассматриваемых тем.

Студент, защитивший все задания расчетно-графической работы, допускается к зачету. Студент, получивший оценку «не зачтено», должен исправить указанные преподавателем ошибки и сдать расчетно-графическую работу на повторную проверку. Студент, не выполнивший расчетно-графическую работу, к зачету или к экзамену не допускается.

2.1.6. Требования к форме представления результатов, оформлению титульного листа и текста расчетно-графической работы

Результаты (задания) расчетно-графической работы оформляются средствами пакета MS Office либо средствами любого графического пакета, которым владеет студент.

Титульный лист оформляют в соответствии с образцом, приведенном в Приложении 2. Титульный лист подписывает автор и руководитель расчетно-графической работы. Фамилии лиц, подписывающих работу, приводятся справа от соответствующих подписей. Перед фамилией руководителя указывают ученое звание и инициалы подписавшего работу.

Расчетно-графическая работа выполняется с применением компьютерных печатающих устройств при использовании текстового редактора Microsoft Office Word.

Работа выполняется на белой бумаге на одной стороне листа А4 (210×297 мм) через 1,5 интервала, шрифтом Times New Roman, 14 пт., форматирование текста по ширине, заголовков — по центру; страница должна иметь поля: левое – 2,5 см, правое – 1,5 см, верхнее – 2 см, нижнее – 2 см. Абзацный отступ – 1,25 см.

Страницы расчетно-графической работы нумеруются арабскими цифрами в правом нижнем углу. На титульном листе и оглавлении цифры не проставляются, хотя они включаются в общую нумерацию страниц.

К оформлению оглавления предъявляются следующие требования: введение и библиографический список не нумеруются.

Не рекомендуется при оформлении текста работы применять несколько различных способов выделения. Следует ограничиться двумя, как правило, это полужирный шрифт и курсив.

Формулы, содержащиеся в расчетно-графической работе, располагают на отдельных строках, выравнивают по центру и нумеруют сквозной нумерацией арабскими цифрами, которые записывают на уровне формулы справа в круглых скобках. Непосредственно под формулой приводится расшифровка символов и

числовых коэффициентов, если они не были пояснены в тексте. В этом случае сразу после формулы (до ее номера) ставится запятая, а первая строка расшифровки (выравнивание по левому краю) начинается словом «где» без двоеточия после него.

Иллюстрации по тексту расчетно-графической работы (рисунки, графики, диаграммы и др.) следует нумеровать арабскими цифрами сквозной нумерацией или нумерацией в пределах главы. Иллюстрации должны быть с подрисовочным текстом. Надписи на иллюстрациях, наименования и подрисовочный текст выполняются шрифтом 12 пт и выравниваются по центру. После наименования рисунка точка не ставится. Перенос части иллюстрации на другую страницу не допускается. Ссылки на иллюстрации в тексте обязательны, они должны связывать иллюстрацию с текстом, при этом должно присутствовать указание на номер (их пишут сокращенно, например: рис. 3). Размещение в тексте иллюстрации не освобождает автора от обязанности пояснить ее содержание.

Таблицы нумеруются арабскими цифрами сквозной нумерацией в пределах всего текста. Слово «Таблица» и порядковый номер помещают над названием таблицы в правом верхнем углу. Если таблица не помещается на одной странице, ее можно продолжить или закончить наследующей, сделав соответствующую надпись – «Продолжение табл.» или «Окончание табл.» (с указанием номера таблицы). Номер таблицы, название и все заполнение выполняется шрифтом 12 пт, интервал между строк минимальный. Ссылки по тексту на таблицы обязательны, их следует приводить в сокращенном виде, например: табл. 4.5. Допускается помещать таблицу вдоль длинной стороны листа (альбомный вариант).

РАЗДЕЛ 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Видом промежуточной аттестации студентов, обучающихся по направлению подготовки 230400.62 «Информационные системы и технологии», без профиля является экзамен во 2 и 3 семестрах.

3.1. Список вопросов для подготовки к экзамену

1. Архитектура ОС Linux.
2. Понятие ядра.
3. Файловая система Linux.
4. Понятие файла и каталога.
5. Командные файлы.
6. Программирование на языке shell.
7. Основные команды управления правами доступа к файлам.
8. Переменные среды.
9. Работа в режиме командной строки.
10. Конвейеры.

11. Управление работами.
12. Управление памятью.
13. Управление вводом-выводом.
14. Редактирование в Linux – текстовый редактор vi.
15. Сценарии оболочки shell.
16. Управление процессами.
17. Взаимодействие процессов.
18. Семафоры в ОС Linux.
19. Сигналы в ОС Linux.
20. Системные вызовы в Linux.
21. Запуск системных вызовов из программ, написанных на языках высокого уровня.
22. Системное администрирование.
23. Проблема безопасности.
24. Монтирование файловых систем.
25. Организация сети.
26. Сетевое программирование.
27. Маршрутизаторы.
28. Интернет-приложения.
29. Распределенная обработка в ОС Linux.
30. Система зависимости файлов ОС Linux.
31. Система поддержки архивов в ОС Linux.
32. Система управления исходным кодом в ОС Linux.
33. Ограничения доступа к SCCS – файлам.
34. Профайлер в ОС Linux.
35. Отладчик в ОС Linux.
36. Методы проектирования базовых и прикладных информационных технологий
37. Методы разработки средств реализации информационных технологий
38. Системные вызовы и библиотечные процедуры.
39. Управление устройствами.
40. Обработка сигналов.
41. Группы процессов и терминалы управления.
42. Таблица процессов.
43. Планировщик. Правила планирования.
44. Таблицы переключения.
45. Сетевой интерфейс.
46. Конфигурирование ядра.
47. Сокеты.

3.2. Общие положения проведения экзамена

К экзамену допускаются студенты, выполнившие в полном объеме график учебного процесса по дисциплине «Системное программирование»: задания лабораторных работ, защитившие расчетно-графические работы, прошедшие тестирование по темам дисциплины согласно Рабочей программе.

Экзаменационная оценка является итоговой по дисциплине и проставляется в приложение к диплому (выписке из зачетной книжки).

ВАРИАНТЫ

РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЫ № 1 «Реализация системных вызовов на языке C в ОС Linux»

Вариант 1

Разработайте программу на языке Си, которая порождает бы процесс потомок, и убедитесь в том, что все файлы, открытые процессом предком, открыты и для процесса потомка.

Вариант 2

Разработайте программу на языке Си для подсчета частоты вхождения слова в состав некоторого текстового файла для каждого хранящегося в этом файле слова.

Вариант 3

Разработайте программу на языке Си с использованием системного вызова *sort*, которая сортирует файл, указанный в командной строке.

Вариант 4

Разработайте программу на языке Си, которая создает текстовый файл и производит запись в него через четыре различных дескриптора файла.

Вариант 5

Разработайте программу на языке Си, которая берет единственный целочисленный аргумент *n* из командной строки и создает двоичное дерево процессов глубины *n*. Когда дерево создано, каждый процесс должен вывести фразу «Я процесс *x*». Узлы дерева процесса должны перечисляться согласно обходу в ширину.

Вариант 6

Разработайте программу на языке Си, которая запрашивает ввод с терминала оператора *i*, если оператор не вводит ответ в течение 10 с, выполняет об-

работку, заданную по умолчанию.

Вариант 7

Разработайте программу на языке Си, которая защищает себя против сигналов *SIGINT*.

Вариант 8

Разработайте программу на языке Си, которая ловит все посланные ей сигналы и распечатывает сведения об этих сигналах. Затем пошлите команду *kill -9* процессу. Как сигнал *SIGKILL* отличается от других сигналов?

Вариант 9

Разработайте программу на языке Си, которая создает кольцо из трех процессов, связанных каналами. Первый процесс должен предложить пользователю ввести строку, а затем послать ее второму процессу. Второй процесс должен реверсировать строку и передать ее третьему процессу. Третий процесс обязан конвертировать строку в прописные буквы и послать ее назад первому процессу. Когда первый процесс получит обработанную строку, он должен вывести ее на терминал. Когда это сделано, все три процесса должны закончиться.

Вариант 10

Разработайте программу на языке Си, которая применяет канал, чтобы позволить родителю читать сообщение от своего дочернего процесса.

ВАРИАНТЫ

РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЫ № 2 «Работа с сигналами»

Вариант 1

Напишите программу, которая вызывает появление сигнала SIGPIPE, когда производится обращение к системному вызову write. Затем заставьте программу игнорировать сигнал SIGPIPE и посмотрите, что возвращает write в этом случае.

Вариант 2

Реализуйте системный вызов pause в терминах sigwait и любого другого системного вызова, какой только захотите.

Вариант 3

Реализуйте системный вызов signal с помощью sigaction.

Вариант 4

Реализуйте системный вызов sigset с помощью sigaction.

Вариант 5

Напишите обработчик сигнала, который выполняет дальний переход по прибытии сигнала и убедитесь, что исполнение программы продолжается с точки вызова setjmp или sigsetjmp.

Вариант 6

Реализуйте системный вызов sleep с помощью usleep, учитывая, что usleep не может выполнять задержку дольше, чем на 1 секунду. То есть вы должны предусмотреть возможность организации более длительных задержек, например на 3 секунды.

Вариант 7

Используя любые системные вызовы, реализуйте команду с именем alarmclock, которая принимает в качестве аргументов время и текст сообщения,

а по достижении заданного времени выводит сообщение на устройство вывода и подает звуковой сигнал.

Вариант 8

Напишите программу для организации двусторонней поочередной связи процесса-родителя и процесса-ребенка через pipe, используя для синхронизации сигналы SIGUSR1 и SIGUSR2.

Вариант 9

Напишите следующие программы. Первая из этих программ посылает пять текстовых сообщений с типом 1 и одно сообщение нулевой длины с типом 255 второй программе. Вторая программа в цикле принимает сообщения любого типа в порядке FIFO и печатает их содержимое до тех пор, пока не получит сообщение с типом 255. Сообщение с типом 255 служит для нее сигналом к завершению работы и ликвидации очереди сообщений. Если перед запуском любой из программ очередь сообщений еще отсутствовала в системе, то программа создаст ее.

Вариант 10

Напишите программу, которая блокирует сигнал SIGINT на время выполнения функции printf.

Типовая форма титульного листа расчетно-графической работы

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ЭКОНОМИКИ И УПРАВЛЕНИЯ «НИНХ»

Институт Прикладной информатики

Кафедра Прикладных информационных технологий

РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА

«Реализация системных вызовов на языке C в ОС Linux»

Учебная дисциплина: Системное программирование

Наименование направления: 230400.62 «Информационные системы и технологии», без профиля

Ф.И.О студента: _____

Номер группы: _____

Номер зачетной книжки: _____

Проверил: _____

Оценочное заключение:

Новосибирск 2011