

Программа учебной дисциплины Операционные системы

Утверждена

Академическим советом ООП

Протокол № 1 от «31» августа 2018 г.

Автор	Истратов Анатолий Юрьевич
Число кредитов	11
Контактная работа (час.)	174
Самостоятельная работа (час.)	244
Курс	4
Формат изучения дисциплины	без использования онлайн курса

I. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

Дисциплина "Операционные системы" имеет целью обучить студентов принципам организации современных вычислительных систем.

Основу курса составляет систематическое изложение теоретических и практических вопросов построения современных операционных систем, концепций и алгоритмов управления локальными и распределенными ресурсами. Изучаются варианты реализации многозадачной и многонитевой обработки (multithreading), организация виртуальной памяти, средства синхронизации, удаленные вызовы процедур (RPC), транзакции, механизмы репликации, средства файловой системы, логическая и физическая реализация файловых систем и систем управления вводом-выводом, подходы к обеспечению безопасности. Обсуждаются современные технологии проектирования ОС: микроядра, модель клиент-сервер, множественные прикладные среды, объектно-ориентированный подход, технология распределенных вычислительных сред (DCE). Особое внимание уделено сетевым возможностям ОС – использованию стандартных протоколов и интерфейсов, возможностям их мультиплексирования.

В рамках курса лекций студенты познакомятся с методологически переработанным материалом, позволяющим эффективно использовать возможности ОС при проектировании самостоятельных прикладных приложений. Студенты не только получают представление о состоянии и возможностях современных ОС, но и приобретут навыки разработки системных приложений.

Изложение принципов организации ОС сопровождается примерами их реализации в конкретных системах: UNIX-подобных, NetWare, Windows 2000 и выше. Особое внимание уделяется UNIX – подобным ОС. Приводится краткое введение в основы ОС UNIX (включая базовые понятия процессов/поток, команд, файлов и т.д.), рассматриваются принципы архитектурной организации системы, основы мобильного программирования и принципы открытых систем, базисный механизм межпроцессных взаимодействий в централизованных и распределенных системах, основные подходы к организации интерактивного взаимодействия с пользователем, вопросы стандартизации. Приводится информация о современном состоянии некоторых коммерческих и свободно распространяемых версий ОС UNIX.

Обсуждаются вопросы взаимодействия различных ОС в рамках одной сети. Изучаются различные подходы к администрированию ОС в больших сетях: доменный подход и подход, основанный на службе каталогов.

В результате освоения дисциплины студенты должны:

- *получить* представление о возможностях и сферах применения современных операционных систем;
- *знать* основные подходы проектирования и организации современных ОС;
- *уметь* работать и разрабатывать приложения в любой современной ОС;
- *уметь* осуществлять взаимодействие между процессами и обрабатывать прерывания;
- *уметь* программировать на командном языке и выполнять административные функции в системе;
- *обладать* навыками проектирования системных приложений и организации межсетевого обмена информацией.

Настоящая дисциплина относится к блоку дисциплин профессионального цикла подготовки – С.З.Б (ФГОС рег.ном. 69 от 17.01.2011)

Изучение дисциплины базируется на дисциплинах естественнонаучного и гуманитарного циклов и в свою очередь обеспечивает изучение следующих дисциплин: «Вычислительные сети», «Системы и сети передачи информации», «Технические средства и методы защиты информации».

Для освоения учебной дисциплины, студенты должны изучить следующие учебные курсы:

- Аппаратные средства вычислительной техники;
- Языки программирования.

II. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Тема 1	Понятие ОС. Аппаратные и программные средства.
Тема 2	Системное окружение
Тема 3	Ядро ОС. Схемы прерываний.
Тема 4	Управление процессами.
Тема 5	Управление памятью. Концепция виртуальной памяти.
Тема 6	Файловая система. Средства файловой системы. Топология файловой системы. Дескриптор файла. Реализация файлов и каталогов
Тема 7	Система управления вводом-выводом. Структурные, аппаратные и программные особенности.
Тема 8	Организация локальных сетей ЭВМ. Основные аппаратные и программные компоненты. Топологии ЛВС.
Тема 9	Операционная система UNIX. Основные характеристики.
Тема 10	Ядро и процессы. Структура процесса ОС UNIX.
Тема 11	Межпроцессные взаимодействия. Управление процессами. Системные вызовы и API.
Тема 12	Файловая система ОС UNIX.
Тема 13	Система управления вводом-выводом (СУВВ) ОС UNIX..
Тема 14	Обработка прерываний в ОС UNIX. Системный вызов «сигнал».
Тема 15	Функционирование ОС UNIX.
Тема 16	Интерпретатор команд SHELL. Программирование на командном языке.
Тема 17	Поддержка в ОС UNIX сетевого обмена.
Тема 18	Взаимодействие с пользователями в рамках системы UNIX.
Тема 19	Примеры реализаций ОС UNIX. SOLARIS, UNIX HP, FREE BSD, MAC, LINUX.

- Тема 20 Межпроцессное взаимодействие на уровне «клиент-сервер» ОС UNIX.
Тема 21 Операционная система WINDOWS. Структура системы.
Тема 22 Уровень аппаратных абстракций ОС WINDOWS.
Тема 23 Уровень ядра ОС WINDOWS.
Тема 24 Исполняющая система ОС WINDOWS.
Тема 25 Реализация объектов в ОС WINDOWS.
Тема 26 Подсистемы окружения ОС WINDOWS.
Тема 27 Процессы и потоки в ОС WINDOWS.
Тема 28 Вызовы для управления заданиями, процессами, потоками и волокнами в ОС WINDOWS.
Тема 29 Межпроцессные взаимодействия в ОС WINDOWS.
Тема 30 Реализация процессов и потоков в ОС WINDOWS.
Тема 31 Планирование в ОС WINDOWS.
Тема 32 Загрузка ОС WINDOWS.
Тема 33 Управление оперативной памятью в ОС WINDOWS.
Тема 34 Управление файловой системой в ОС WINDOWS.

III. ОЦЕНИВАНИЕ

Преподаватель оценивает работу студентов на лекциях и практических занятиях: активность при ответе на вопросы преподавателя, правильность выполнения заданий на лабораторных работах. Оценки за работу на лекциях и практических занятиях преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за работу на лекционных и практических занятиях определяется перед промежуточным или итоговым контролем - $O_{аудиторная}$.

Преподаватель оценивает самостоятельную работу студентов: правильность выполнения домашних работ. Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за самостоятельную работу определяется перед промежуточным или итоговым контролем – $O_{сам. работа}$.

Преподаватель оценивает выполнение контрольных работ по 10-ти балльной шкале – $O_{контр. раб.}$

Накопленная оценка за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

$$O_{текущий} = O_{к/р} ;$$

Способ округления накопленной оценки текущего контроля: в пользу студента.

Результирующая оценка за итоговый контроль в форме экзамена(зачета) выставляется по следующей формуле, где $O_{экз}$ – оценка за работу непосредственно на экзамене(зачете):

$$O_{итоговый} = 0,4 \cdot O_{экз} + 0,1 \cdot O_{текущий} + 0,2 \cdot O_{сам. работа} + 0,1 \cdot O_{аудиторная} + 0,2 O_{контр. раб.}$$

Способ округления накопленной оценки итогового контроля в форме экзамена: в пользу студента.

На передаче студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

На экзамене студент может получить дополнительную практическую задачу, которая оценивается в 1 балл. Таким образом, результирующая оценка за итоговый контроль в форме экзамена(зачета) получаемая на передаче, выставляется по формуле:

$$O_{\text{итоговый}} = (0,4 \cdot O_{\text{ЭКЗ}} + 0,1 \cdot O_{\text{текущий}} + 0,2 \cdot O_{\text{сам. работа}} + 0,1 \cdot O_{\text{аудиторная}} + 0,2 O_{\text{контр. раб.}}) + O_{\text{доп. вопрос}}$$

В диплом выставляет результирующая оценка по учебной дисциплине, которая формируется по следующей формуле:

$$O_{\text{дисциплина}} = O_{\text{итоговый}}$$

Способ округления результирующей оценки по учебной дисциплине: в пользу студента.

В диплом ставится оценка за итоговый контроль, которая является результирующей оценкой по учебной дисциплине.

IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

Тематика контрольных работ.

1. Определение возвращаемых числовых значений дескрипторов файлов и пересылаемых байтов ввода/вывода по заданным фрагментам программ на языке программирования C/C++ в ОС UNIX.
2. Разработка командных файлов на языке программирования SHELL в ОС UNIX.
3. Вызовы API Win32 для обмена информацией между параллельно работающими процессами/потоками.
4. Управление приоритетами параллельно работающих потоков в ОС WINDOWS.

Тематика домашних заданий

1. Раскрыть темы:

- Расслоение памяти
- Регистр перемещений
- Прерывания и опрос состояний
- Буферизация
- Периферийные устройства
- Защита памяти
- Таймер и часы
- Каналы ввода-вывода
- Захват цикла памяти
- Относительная адресация
- Режимы работы компьютера
- Виртуальная память
- Мультипроцессорная обработка
- Прямой доступ к памяти
- Конвейеризация

- Иерархия памяти
- Программирование на машинном языке
- Ассемблеры и макропроцессоры
- Компиляторы
- Интерпретаторы
- Процедурно-ориентированные и проблемно-ориентированные языки
- Спулинг
- Абсолютные и перемещающие загрузчики
- Связывающие загрузчики и редакторы связей
- Микропрограммы
- Горизонтальный и вертикальный микрокод
- Эмуляция

2. Расписать назначение, аргументы и возвращаемые значения следующих системных вызовов (API) UNIX-подобных ОС:

- | | | |
|-----------|-----------------------------|------------------|
| • alarm | • pause | • getitimer |
| • chdir | • profil | • msgget |
| • chroot | • ptrace | • msgsnd |
| • chmod | • setsid | • msgrcv |
| • fchmod | • setpgid | • msgctl |
| • chown | • setuid | • semget |
| • fchown | • setgid | • semop |
| • lchown | • seteuid | • semctl |
| • access | • setegid | • shmget |
| • brk | • signal, sigset, sigaction | • shmat |
| • sbrk | (привести таблицу | • shmdt |
| • exit | сигналов с действиями по | • shmctl |
| • _exit | умолчанию и причинами | • socket |
| • getpid | посылки сигнала) | • bind |
| • getppid | • stat | • listen |
| • getuid | • fstat | • accept |
| • geteuid | • stime | • connect |
| • getegid | • symlink | • send, sendto |
| • getgid | • sync | • recv, recvfrom |
| • ioctl | • time | • shutdown |
| • indir | • ftime | • mmap |
| • kill | • times | • munmap |
| • umask | • vfork | • msync |
| • link | • waitpid | • opendir |
| • unlink | • execlp | • closedir |
| • lseek | • execl | • readdir |
| • mknod | • execvp | • rewinddir |
| • mkdir | • execv | • scanfdir |
| • rmdir | • execve | • cfsetospeed |
| • mkfifo | • popen | • cfsetispeed |
| • mount | • pclose | • cfgetospeed |

- umount
- nice
- sigprocmask
- sigpending
- setitimer
- cfgetispeed
- tcsetattr
- getattr

3. Расписать назначение и опции (ключи) следующих команд файловой системы UNIX-подобных ОС:

- pwd
- ls
- cat
- cp
- mv
- pr
- lpr
- lpq
- who
- ps
- tty
- du
- cal
- cd
- mkdir
- rmdir
- rm
- chmod
- chown
- chgrp
- cmp
- find
- file
- ar
- ranlib
- tee
- sort
- cut
- head
- tail
- spell
- grep
- fgrep
- more
- wc

Тематика лабораторных работ

№ п/п	№ раздела дисциплины	Наименование лабораторных работ
1	11	Обмен информацией между параллельными процессами в ОС UNIX.
2	12	Организация защиты файлов в файловой системе UNIX,
3	14	Обработка прерываний в ОС UNIX.
4	16	Программирование на командном языке.
5-7	20	Межпроцессное взаимодействие на уровне «клиент-сервер».

Варианты лабораторных работ

1. Варианты к лабораторным работам 1, 2, 3, 4 содержатся в источнике: Истратов А.Ю. - Программирование в операционной среде UNIX: обмен информацией между параллельными процессами, организация защиты файлов в файловой системе, обработка прерываний (Учебное пособие), М.: РГУИТиП, 2006 г.
2. Варианты к лабораторной работе № 5 «Обмен сообщениями между асинхронными процессами»:

Вариант № 1

Клиент. Создать очередь сообщений. Передать в эту очередь ряд сообщений, содержащих: а). имена пользователей, посылавшим Вам электронную почту в течении последней недели; б). имена терминалов, с которых посылалась электронная почта; в). Количество отправленных Вам почтовых сообщений.

Сервер. Выбрать из очереди сообщений, созданной сервером, последнее сообщение указанного типа. Записать в стандартный файл вывода это сообщение и общее число сообщений в очереди. После этого удалить очередь сообщений.

Вариант №2

Сервер. Создать очередь сообщений. Записать в качестве 1-ого сообщения имена всех текстовых файлов текущего каталога, в качестве 2-ого сообщения – имена всех файлов текущего каталога, написанных на языке программирования Си, в качестве 3-его сообщения – имена всех двоичных файлов текущего каталога.

Клиент 1. Прочитать 2-ое сообщение из очереди и определить количество строк в каждом файле.

Клиент 2. Определить время, когда последнее сообщение было записано в очередь и вывести его на экран

Вариант №3

Клиент. Создать очередь сообщений. Передать в эту очередь информацию (имена) о текстовых файлах текущего каталога. Вывести на экран ответы сервера.

Сервер. Выбрать из очереди самое старое сообщение указанного типа. Определить те текстовые файлы, количество строк в которых превышает 10, и послать об этом сообщение клиенту. Определить время, когда в очередь было передано самое последнее сообщение.

Вариант №4

Клиент. Создать очередь сообщений. Передать в эту очередь список активных процессов, присутствующих в системе, не управляемых терминалами.

Сервер. Выбрать из очереди сообщений, созданной клиентом, последнее сообщение. Определить в полученном списке процессов те процессы, которые находятся в оперативной памяти, а также общее число сообщений в очереди. Записать в стандартный файл вывода эти данные, после чего удалить очередь сообщений.

Вариант №5

Клиент. Создать очередь сообщений. Передать в эту очередь полное имя текущего каталога и список файлов текущего каталога, в которых встречается подстрока «define».

Сервер. Выбрать из очереди все сообщения. Отсортировать список файлов из очереди по времени создания и записать эту информацию в стандартный файл вывода. Определить идентификатор процесса, который последним передал сообщение в очередь и максимальную длину очереди сообщений в байтах. Удалить очередь сообщений.

Вариант №6

Сервер. Создать очередь сообщений. Записать в неё сообщение об именах файлов, помещенных в спулинг.

Клиент. Выбрать из очереди сообщений последнее сообщение. Отфильтровать в нем те файлы, последняя модификация которых производилась в течение текущего дня и

записать их в стандартный файл вывода. Вывести также величину общего количества байтов во всех сообщениях очереди. Удалить очередь сообщений.

Вариант №7

Сервер. Создать очередь сообщений. Записать в качестве 1-го сообщения количество пользователей, работающих в настоящее время в системе, в качестве 2-ого сообщения – имена этих пользователей, в качестве 3-его сообщения – суммарное рабочее время, затраченное пользователями, в качестве 4-ого сообщения – имена задействованных терминалов.

Клиент 1. Определить общее число байтов во всех сообщениях очереди.

Клиент 2. Определить идентификатор процесса, который последним передал в очередь сообщение.

Клиент 3. Прочитать 3-е сообщение из очереди и вывести его на печать.

Вариант №8

Сервер. Создать очередь сообщений. Записать в качестве 1-го сообщения текущую дату и время, в качестве 2-го сообщения – имена всех пользователей, работающих в настоящее время в системе, в качестве 3-его сообщения – хост-имя компьютера. Сообщения, полученные от клиентов, распечатывать.

Клиент 1. Выбрать из очереди сообщений 2-ое сообщение и определить имена терминалов, связанных с этим пользователем.

Клиент 2. Выбрать из очереди 1-ое сообщение, определить по нему день недели, а также записать эту информацию в очередь сообщений.

Вариант №9

Клиент. Создать очередь сообщений. Записать в качестве 1-ого сообщения количество активных процессов, работающих в настоящее время в системе и управляемых терминалами, в качестве 2-го сообщения – значение приоритетов этих процессов.

Сервер. Выбрать из очереди 2-ое сообщение и определить процесс с наивысшим приоритетом, а также имя владельца этого процесса и время, когда это сообщение было записано в очередь.

Вариант № 10

Сервер. Создать очередь сообщений. Записать в качестве 1-ого сообщения имена всех файлов текущего каталога, содержащих в имени латинскую букву «а», в качестве 2-го сообщения даты последней модификации этих файлов.

Клиент. Определить из информации, записанной в сообщениях очереди, количество файлов, размер которых не превышает 2-х блоков и имя файла с самой старой модификацией, а также время, когда из очереди было прочитано самое последнее сообщение.

Вариант №11

Клиент. Создать очередь сообщений. Записать в качестве 1-ого сообщения имена всех командных файлов текущего каталога, в качестве 2-го сообщения – общее количество строк этих файлов, в качестве 3-его сообщения – суммарный размер этих файлов в байтах.

Сервер. Выбрать из очереди 1-ое сообщение. Определить все файлы из сообщения, в которых используется фильтр «awk», подсчитать их количество и определить максимальное число байтов всех сообщений очереди.

Вариант №12

Сервер. Создать очередь сообщений. При поступлении клиентского сообщения о календаре текущего месяца, определить текущий день недели по текущей дате, а также идентификатор процесса, который записал сообщение в очередь.

Клиент. Переслать в очередь сообщение, содержащее календарь текущего месяца.

Вариант №13

Сервер. Создать очередь сообщений. Записать в неё сообщение об именах файлов, содержащих командный текст.

Клиент. Выбрать из очереди сообщений последнее сообщение. Отфильтровать в нем те файлы, в которых осуществляется обработка прерываний и записать их в стандартный файл вывода. Вывести также значение идентификатора процесса, который последним передал сообщение в очередь. Удалить очередь сообщений.

Вариант №14

Сервер. Создать очередь сообщений. Записать в качестве 1-го сообщения все поддиректории текущего каталога, в качестве 2-го сообщения – количество, находящихся в них файлов, в качестве 3-его сообщения – хост-имя компьютера. Сообщения, полученные от клиентов, распечатывать. Удалить очередь сообщений.

Клиент 1. Выбрать из очереди сообщений 2-ое сообщение и определить имена владельцев этих файлов.

Клиент 2. Выбрать из очереди 1-ое сообщение, определить размер в байтах каждой поддиректории и записать эту информацию в очередь сообщений.

Вариант №15

Клиент. Создать очередь сообщений. Записать в качестве 1-ого сообщения имена всех файлов текущего каталога представленных на машинном языке, в качестве 2-го сообщения – суммарный размер этих файлов в байтах.

Сервер. Выбрать из очереди 1-ое сообщение и определить файл максимального размера, а также имя владельца этого файла, время последней модификации и время, когда сообщение было записано в очередь. Удалить очередь сообщений.

Вариант №16

Клиент. Создать очередь сообщений. Передать в эту очередь информацию (имена) о файлах текущего каталога, написанных на языке программирования Си. Вывести на экран ответы сервера и удалить очередь сообщений.

Сервер. Выбрать из очереди самое старое сообщение. Определить те файлы, в которых встречается конструкция цикла, и послать об этом сообщение клиенту. Определить время, когда из очереди было прочитано самое последнее сообщение.

Вариант №17

Сервер. Создать очередь сообщений. Записать в качестве 1-го сообщения идентификаторы всех процессов, принадлежащих данному пользователю, в качестве 2-ого сообщения – приоритеты этих процессов, в качестве 3-его сообщения – суммарное рабочее время, затраченное этими процессами, в качестве 4-ого сообщения – имена задействованных терминалов.

Клиент 1. Определить общее число байтов во всех сообщениях очереди.

Клиент 2. Определить идентификатор процесса, который последним передал в очередь сообщение.

Клиент 3. Прочитать 3-е сообщение из очереди и вывести его на печать. Удалить очередь сообщений

Вариант №18

Клиент. Создать очередь сообщений. Передать в эту очередь имя файла текущего каталога, который был модифицирован последним.

Сервер. Выбрать из очереди сообщений, созданной клиентом, последнее сообщение. Определить количество строк указанного файла, а также общее число сообщений в очереди. Записать в стандартный файл вывода эти данные, после чего удалить очередь сообщений.

Вариант №19

Сервер. Создать очередь сообщений. Записать в качестве 1-ого сообщения имена всех файлов текущего каталога, размер которых превышает 50 байтов, в качестве 2-ого сообщения – имена всех файлов текущего каталога, написанных на языке программирования Си, в качестве 3-его сообщения – имена всех файлов текущего каталога, созданных в последние 2 дня.

Клиент 1. Прочитать 2-ое сообщение из очереди и определить количество строк в каждом файле.

Клиент 2. Определить время, когда последнее сообщение было записано в очередь и вывести его на экран монитора, после чего удалить очередь сообщений.

Вариант №20

Сервер. Создать очередь сообщений. Записать в неё сообщение об именах файлов файловой системы, принадлежащих данному пользователю.

Клиент. Выбрать из очереди сообщений последнее сообщение. Отфильтровать в нём те файлы, последняя модификация которых производилась в течение текущего дня и записать их в стандартный файл вывода. Вывести также величину общего количества байтов во всех сообщениях очереди. Удалить очередь сообщений.

Вариант №21

Сервер. Создать очередь сообщений. Записать в качестве 1-ого сообщения имена всех процессов, приоритет которых больше 30, в качестве 2-го сообщения время работы этих процессов.

Клиент. Определить из информации, записанной в сообщениях очереди, общее количество выделенных процессов и идентификатор самого «старого» работающего процесса, а также время, когда из очереди было прочитано самое последнее сообщение. Удалить очередь сообщений.

Вариант №22

Клиент. Создать очередь сообщений. Передать в эту очередь информацию об очередях сообщений, присутствующих в системе, а также наборах семафоров, если они есть.

Сервер. Выбрать из очереди все сообщения. Отсортировать список данных из очереди по времени создания и записать эту информацию в стандартный файл вывода. Определить идентификатор процесса, который последним передал сообщение в очередь и максимальную длину очереди сообщений в байтах. Удалить очередь сообщений.

Список источников

1. Чан Т. Системное программирование на C++ для UNIX. – Киев: Издательская группа BHV, 1999 г.
2. Моли Б. UNIX/LINUX: Теория и практика программирования. - М: КУДИЦ_ОБРАЗ, 2004 г.
3. Роббинс А. LINUX: программирование в примерах. – М: КУДИЦ_ОБРАЗ, 2005 г.

4. Истратов А.Ю. Межпроцессное взаимодействие на уровне «клиент-сервер» в ОС UNIX (Учебное пособие), М.: РГУИТиП, 2006 г.

3. Варианты к лабораторной работе № 6 «Обмен информацией через разделяемую память между асинхронными процессами»:

Вариант № 1

Сервер. Создать разделяемую область памяти (РОП). Подсоединить её к виртуальному адресному пространству процесса. Записать в неё информацию обо всех файлах текущего каталога. После того как клиент прочтет информацию из РОП удалить ее.

Клиент. Открыть разделяемую область памяти, созданную сервером. Прочитать записанные сообщения и записать в стандартный файл вывода информацию только о двоичных файлах. Вывести также идентификатор процесса, который последним подсоединился к разделяемой области памяти.

Вариант №2

Сервер. Создать набор семафоров и разделяемую область памяти (РОП). Из информации, переданной клиентом, выделить имена тех пользователей, которые работают в системе более 20 минут, а также время, когда какой-либо процесс последним изменял управляющие параметры РОП. Удалить РОП и набор семафоров.

Клиент. Записать, в созданную сервером разделяемую область памяти, список всех пользователей, работающих в настоящее время в системе.

Вариант №3

Сервер. Создать набор семафоров и разделяемую область памяти (РОП). Подождать, пока клиент не пришлет информацию. Определить права владельца каждого файла и переслать эти данные через разделяемую область памяти клиенту.

Клиент. Записать в созданную сервером разделяемую область памяти информацию (имена) обо всех файлах текущего каталога. После того, как будет получена информация о владельцах файлов, вывести время, когда процесс последний раз подключался к разделяемой области памяти. Удалить РОП и набор семафоров.

Вариант №4

Сервер. Создать разделяемую область памяти. Подсоединить её к виртуальному адресному пространству процесса. Записать в неё информацию (идентификаторы) обо всех активных процессах в системе, управляемых терминалами. После обработки информации клиентом удалить разделяемую область памяти

Клиент. Открыть разделяемую область памяти, созданную серверным процессом. Считать из неё информацию. Определить состояния процессов, идентификаторы которых указаны в сообщении. Определить также количество процессов, подсоединенных в данный момент времени к разделяемой области памяти.

Вариант №5

Сервер. Создать разделяемую область памяти и набор семафоров. Подождать, пока один из клиентов не пришлет информацию. Среди полученных имен файлов, определить такие, размер которых превышает 2 блока, и эти данные переслать через разделяемую область памяти соответствующему клиенту. После обработки информации клиентами удалить РОП и НС.

Клиент 1. Записать в разделяемую область памяти, созданную сервером, имена текстовых файлов текущего каталога. Записать в стандартный файл вывода результаты обработки сервера и время последнего отсоединения процесса от РОП.

Клиент 2. Записать в разделяемую область памяти, созданную сервером, имена файлов текущего каталога, написанные на языке программирования Си. Записать в стандартный файл вывода результаты обработки этого запроса.

Вариант №6

Сервер. Создать набор семафоров и разделяемую область памяти. Вывести информацию, полученную от клиентов в стандартный файл вывода. После этого вывести все значения набора семафоров и удалить РОП и НС.

Клиент 1. Подсоединиться к разделяемой области памяти. Определить подкаталог текущего каталога с максимальным количеством файлов и записать эту информацию в разделяемую область памяти.

Клиент 2. Подсоединиться к разделяемой области памяти. Определить количество процессов, подсоединенных к разделяемой области памяти и записать эту информацию в неё.

Вариант №7

Сервер. Создать разделяемую область памяти и набор семафоров. Ждать сообщений клиентов. При поступлении сообщения от 1-ого клиента, обработать его и переслать абоненту размеры полученных файлов. Распечатать показания времени, а также содержимое спулинга, полученное от 2-ого клиента. Удалить РОП и НС.

Клиент 1. Записать в разделяемую область памяти имена тех файлов текущего каталога, в которых встречается строка с шаблоном «main», а также время, когда какой-либо процесс последний раз выполнял операцию над семафорами набора. Распечатать ответ сервера.

Клиент 2. Записать в разделяемую область памяти содержимое спулинга.

Вариант №8

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров(НС). Записать в РОП текущую дату и время, имена всех пользователей, работающих в настоящее время в системе и хост-имя компьютера. Сообщения, полученные от клиентов, распечатывать. В конце сеанса удалить РОП и НС.

Клиент 1. Выбрать из РОП имена всех пользователей, работающих в настоящее время в системе и определить имена терминалов, связанные с этими пользователями.

Клиент 2. Выбрать из РОП текущую дату и время и определить по ним день недели, а также количество семафоров в наборе.

Вариант №9

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). При поступлении клиентского сообщения в РОП определить системную составляющую приоритета полученных процессов, а также время когда последний процесс выполнял операции над семафорами и переслать клиенту. Удалить РОП и НС.

Клиент. Подсоединиться к РОП. Определить идентификаторы процессов, являющихся системными и переслать эти данные в серверное гнездо. При получении ответа от сервера, распечатать поступившую информацию.

Вариант № 10

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). Среди поступивших в РОП от клиентов сообщений, отфильтровать такие, которые содержат

шаблон «S» и вывести их на печать, а также вывести значение идентификатора. Удалить РОП и НС.

Клиент 1. Записать в РОП, созданную сервером сообщение, содержащее полную информацию о списке всех активных процессов в системе неуправляемых терминалами.

Клиент 2. Записать в РОП, созданную сервером сообщение, содержащее полную информацию о списке всех активных процессов в системе управляемых терминалами, а также идентификатор процесса создателя РОП.

Вариант №11

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). При поступлении клиентского сообщения о календаре текущего месяца, определить текущий день недели по текущей дате, а также идентификатор процесса который последним отсоединялся от РОП. Удалить РОП и НС.

Клиент. Записать в РОП, созданную сервером сообщение, содержащее календарь текущего месяца.

Вариант №12

Сервер. Создать набор семафоров и разделяемую область памяти (РОП). Подождать, пока клиент не пришлет информацию. Определить права владельца каждой РОП и переслать эти данные через разделяемую область памяти клиенту.

Клиент. Записать в созданную сервером разделяемую область памяти информацию обо всех РОП системы. После того, как будет получена информация о владельцах РОП, вывести время, когда процесс последний раз подключался к разделяемой области памяти. Удалить РОП и набор семафоров.

Вариант №13

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). При поступлении клиентского сообщения в РОП определить время работы каждого пользователя, а также время когда последний процесс выполнял операции над семафорами и переслать клиенту. Удалить РОП и НС.

Клиент. Подсоединиться к РОП. Определить имена пользователей, работающих в системе и переслать эти данные в серверное гнездо. При получении ответа от сервера, распечатать поступившую информацию.

Вариант №14

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров(НС). Ждать сообщений клиентов. При поступлении сообщения от 1-ого клиента, обработать его и переслать абоненту имена «родителей» этих процессов. Распечатать показания времени, а также содержимое сообщения, полученное от 2-ого клиента. Удалить РОП и НС.

Клиент 1. Записать в разделяемую область памяти идентификаторы всех процессов, приоритет которых больше 25, а также время работы этих процессов. Распечатать ответ сервера.

Клиент 2. Записать в разделяемую область памяти время, когда какой-либо процесс последний раз выполнял операцию над семафорами набора.

Вариант №15

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров(НС). Подождать, пока один из клиентов не пришлет информацию. Среди полученных имен файлов, определить такие, в которых больше 4 строк, и эти данные переслать через разделяемую область памяти соответствующему клиенту. После обработки информации клиентами удалить РОП и НС.

Клиент 1. Записать в разделяемую область памяти, созданную сервером, имена всех файлов текущего каталога, размер которых превышает 80 байтов. Записать в стандартный файл вывода результаты обработки сервера и время последнего отсоединения процесса от РОП.

Клиент 2. Записать в разделяемую область памяти, созданную сервером, имена файлов текущего каталога, написанные на командном языке программирования. Записать в стандартный файл вывода результаты обработки этого запроса.

Вариант №16

Сервер. Создать набор семафоров (НС) и разделяемую область памяти (РОП). Подождать, пока клиент не пришлет информацию. Определить права владельца каждого файла и переслать эти данные через разделяемую область памяти клиенту.

Клиент. Записать в созданную сервером разделяемую область памяти информацию (имена) обо всех файлах файловой системы, принадлежащих данному пользователю. После того, как будет получена информация о правах владельца файлов, вывести время, когда процесс последний раз подключался к разделяемой области памяти. Удалить РОП и набор семафоров.

Вариант №17

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). При поступлении клиентского сообщения в РОП определить время нахождения пользователей в системе, а также время когда последний процесс выполнял операции над семафорами и переслать клиенту. Удалить РОП и НС.

Клиент. Подсоединиться к РОП. Определить имена пользователей, работающих в настоящее время в системе и переслать эти данные в серверное гнездо. При получении ответа от сервера, распечатать поступившую информацию.

Вариант №18

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров(НС). Записать в РОП идентификаторы всех процессов, принадлежащих данному пользователю, приоритеты этих процессов, суммарное рабочее время, затраченное этими процессами и имена задействованных терминалов. Сообщения, полученные от клиентов, распечатывать. В конце сеанса удалить РОП и НС.

Клиент 1. Выбрать из РОП идентификатор процесса – «долгожителя» и определить текущее его состояние.

Клиент 2. Выбрать из РОП идентификатор процесса с наименьшим приоритетом, а также определить количество семафоров в наборе.

Вариант №19

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров(НС). Подсоединить РОП к виртуальному адресному пространству процесса. Записать в неё информацию о файлах, содержащихся в спулинге. После обработки информации клиентом удалить разделяемую область памяти

Клиент. Открыть разделяемую область памяти, созданную серверным процессом. Считать из неё информацию. Определить имена файлов, количество строк в которых больше 10. Определить также количество процессов, подсоединенных в данный момент времени к разделяемой области памяти.

Вариант №20

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). От поступивших в РОП от клиентов сообщений, сформировать информацию о количестве всех

файлов, содержащихся в подкаталогах текущего каталога и вывести её на печать, а также вывести значение идентификатора. Удалить РОП и НС.

Клиент 1. Записать в РОП, созданную сервером сообщение, содержащее имена всех подкаталогов текущего каталога.

Клиент 2. Записать в РОП, созданную сервером сообщение, содержащее идентификатор процесса создателя РОП.

Вариант №21

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров (НС). При поступлении клиентского сообщения, определить время создания этих файлов, а также идентификатор процесса который последним отсоединился от РОП. Удалить РОП и НС.

Клиент. Записать в РОП, созданную сервером сообщение, содержащее список всех файлов текущего каталога, представленных на машинном языке.

Вариант №22

Сервер. Создать разделяемую область памяти (РОП) и набор семафоров(НС). Подсоединить РОП к виртуальному адресному пространству процесса. Записать в неё информацию обо всех наборах семафоров, созданных в системе. После того как клиент прочтет информацию из РОП удалить ее и набор семафоров.

Клиент. Открыть разделяемую область памяти, созданную сервером. Прочитать записанные сообщения и записать в стандартный файл вывода информацию только о «чужих» НС. Вывести также идентификатор процесса, который последним подсоединился к разделяемой области памяти.

Список источников

1. Чан Т. Системное программирование на C++ для UNIX. – Киев: Издательская группа ВНУ, 1999 г.
1. Моли Б. UNIX/LINUX: Теория и практика программирования. - М: КУДИЦ_ОБРАЗ, 2004 г.
2. Роббинс А. LINUX: программирование в примерах. – М: КУДИЦ_ОБРАЗ, 2005 г.
3. Стивене Р.У., Раго С.А. UNIX. Профессиональное программирование. 2-е издание. – СПб.: Символ-Плюс, 2007 г.
4. Истратов А.Ю. Межпроцессное взаимодействие на уровне «клиент-сервер» в ОС UNIX (Учебное пособие), М.: РГУИТиП, 2006 г.

4. Варианты к лабораторной работе № 7 «Обмен информацией через гнезда между асинхронными процессами»

Вариант № 1

Сервер. Создать гнездо без установления соединения домена INET. Присвоить ему имя. При поступлении клиентского сообщения определить системную составляющую приоритета полученных процессов и переслать клиенту.

Клиент. Создать гнездо без установления соединения домена INET. Определить идентификаторы процессов, являющихся системными и переслать эти данные в серверное гнездо. При получении ответа от сервера, распечатать поступившую информацию.

Вариант №2

Сервер. Создать гнездо домена UNIX типа virtual circuit. Назначить ему адрес. Среди поступивших от клиентов сообщений, отфильтровать такие, которые содержат шаблон «S».

Клиент 1. Создать гнездо домена UNIX типа virtual circuit. Послать в серверное гнездо сообщение, содержащее полную информацию о списке всех активных процессов в системе не управляемых терминалами.

Клиент 2. Создать гнездо домена UNIX типа virtual circuit. Послать в серверное гнездо сообщение, содержащее полную информацию о списке всех активных процессов в системе управляемых терминалами.

Вариант №3

Сервер. Создать гнездо без установления соединения домена UNIX. Присвоить ему адрес. Послать в клиентское гнездо данные (идентификаторы) обо всех активных процессах системы управляемых терминалами. Результаты обработки клиентом этих данных распечатать.

Клиент. Создать гнездо без установления соединения домена UNIX. Отфильтровать информацию из серверного гнезда с целью выявления тех процессов, которые принадлежат данному пользователю. Результаты обработки передать в серверное гнездо.

Вариант №4

Сервер. Создать гнездо с установлением соединения домена INET. Присвоить ему имя. При поступлении клиентского сообщения о календаре текущего месяца, определить текущий день недели по текущей дате.

Клиент. Создать гнездо с установлением соединения домена INET. Переслать в серверное гнездо сообщение, содержащее календарь текущего месяца.

Вариант №5

Сервер. Создать гнездо без установления соединения домена UNIX. Присвоить ему имя. Записать в него информацию о количестве файлов текущего каталога. Распечатать информацию, полученную от клиента.

Клиент. Создать гнездо без установления соединения домена UNIX. Прочитать сообщение из серверного гнезда. Уменьшить прочитанное значение на число, равное количеству файлов, имя которых начинается с точки и передать это значение в гнездо сервера.

Вариант №6

Сервер. Создать гнездо домена INET типа virtual circuit. Присвоить ему имя. Записать в него информацию, содержащую имена файлов текущего каталога, которые были модифицированы в течении последнего месяца. Распечатать ответ клиента.

Клиент. Создать гнездо домена INET типа virtual circuit. Прочитать сообщение из серверного гнезда. Определить имена файлов, которые нельзя выполнять для данного пользователя и отправить их в виде сообщения в серверное гнездо.

Вариант №7

Сервер. Создать гнездо домена UNIX типа virtual circuit. Присвоить ему имя. Принять сообщение от клиентского гнезда. Распечатать электронную почту 1-го по порядку пользователя, имя которого упоминается в ответе клиента.

Клиент. Создать гнездо домена UNIX типа virtual circuit. Передать в серверное гнездо информацию (имена) обо всех пользователях, от которых получена электронная почта.

Вариант №8

Сервер. Создать гнездо домена UNIX типа datagram. Принять сообщение от клиентского гнезда. Записать в стандартный файл вывода имена тех файлов из принятого сообщения, размер которых не превышает 4 блока.

Клиент. Создать гнездо домена UNIX типа datagram. Передать сообщение в серверное гнездо обо всех файлах текущего каталога, написанных на языке программирования Си.

Вариант №9

Сервер. Создать гнездо с установлением соединения домена INET. Присвоить ему имя. При поступлении клиентского сообщения, определить количество файлов в каждой поддиректории текущего каталога и переслать эту информацию в клиентское гнездо.

Клиент. Создать гнездо с установлением соединения INET. Переслать в серверное гнездо сообщение обо всех поддиректориях текущего каталога. Записать в стандартный файл вывода ответ сервера.

Вариант № 10

Сервер. Создать гнездо без установления соединения домена INET. При поступлении клиентских сообщений подсчитать соотношение между количеством пользователей и задействованными терминалами и переслать результат клиенту.

Клиент. Создать гнездо без установления соединения домена INET. Передать в серверное гнездо ряд сообщений, содержащих: а). имена пользователей, посылавшим Вам электронную почту в течении последней недели; б). имена терминалов, с которых посылалась электронная почта; в). Количество отправленных Вам почтовых сообщений.

Распечатать ответ сервера.

Вариант №11

Сервер. Создать гнездо дейтаграммного типа домена UNIX с установлением соединения. При поступлении клиентского сообщения определить сколько времени выделенные процессы находились в распоряжении центрального процессора и их приоритет.

Клиент. Создать гнездо дейтаграммного типа домена UNIX с установлением соединения. Выявить среди активных процессов, управляемых терминалами, процессы находящиеся в состоянии «S» и переслать эту информацию в серверное гнездо.

Вариант №12

Сервер. Создать гнездо без установления соединения домена UNIX. Присвоить ему имя. Записать в него информацию о количестве текстовых файлов текущего каталога. Распечатать информацию, полученную от клиента.

Клиент. Создать гнездо без установления соединения домена UNIX. Прочитать сообщение из серверного гнезда. Уменьшить прочитанное значение на число, равное количеству командных файлов текущего каталога и передать это значение в гнездо сервера.

Вариант №13

Сервер. Создать гнездо домена UNIX типа virtual circuit. Присвоить ему имя. Принять сообщение от клиентского гнезда. Определить количество пользователей, имена которых упоминаются в ответе клиента.

Клиент. Создать гнездо домена UNIX типа virtual circuit. Передать в серверное гнездо информацию (имена) обо всех пользователях, работающих в настоящее время в системе.

Вариант №14

Сервер. Создать гнездо без установления соединения домена INET. Присвоить ему имя. Записать в гнездо информацию(имена) обо всех файлах текущего каталога. Распечатать ответы клиентов. Клиент1. Создать гнездо без установления соединения домена INET. Считать из серверного гнезда информацию. Определить количество строк, содержащихся в файлах, имена которых указаны в полученной информации, и передать ответ в серверное гнездо.

Клиент1. Создать гнездо без установления соединения домена INET. Считать из серверного гнезда информацию. Определить количество строк, содержащихся в файлах, имена которых указаны в полученной информации, и передать ответ в серверное гнездо.

Клиент2. Создать гнездо без установления соединения домена INET. Считать из серверного гнезда информацию. Определить количество байтов, содержащихся в файлах, имена которых указаны в полученной информации, и передать ответ в серверное гнездо.

Вариант № 15

Сервер. Создать гнездо без установления соединения домена INET. При поступлении клиентского сообщения отсортировать список файлов из клиентского гнезда по времени создания и записать эту информацию в стандартный файл вывода и в клиентское гнездо.

Клиент. Создать гнездо без установления соединения домена INET. Передать в серверное гнездо полное имя текущего каталога и список файлов, в которых встречается подстрока «main». Распечатать ответ сервера.

Вариант №16

Сервер. Создать гнездо домена UNIX типа datagram. Принять сообщение от клиентского гнезда. Определить количество строк указанного файла, а также имя его владельца и группы. Записать результат в стандартный файл вывода.

Клиент. Создать гнездо домена UNIX типа datagram. Передать сообщение в серверное гнездо об имени файла текущего каталога, который был модифицирован последним.

Вариант №17

Сервер. Создать гнездо с установлением соединения домена INET. Присвоить ему имя. При поступлении клиентского сообщения, определить общее количество выделенных процессов и идентификатор самого «молодого» работающего процесса и переслать эту информацию в клиентское гнездо.

Клиент. Создать гнездо с установлением соединения INET. Переслать в серверное гнездо сообщение обо всех процессах, приоритет которых больше 25, а также время работы этих процессов. Записать в стандартный файл вывода ответ сервера.

Вариант №18

Сервер. Создать гнездо домена UNIX типа virtual circuit. Назначить ему адрес. Среди поступивших от клиентов сообщений, отфильтровать такие, которые имеются и в том и в другом списке.

Клиент 1. Создать гнездо домена UNIX типа virtual circuit. Послать в серверное гнездо сообщение, содержащее имена всех файлов текущего каталога, размер которых не превышает 100 байтов.

Клиент 2. Создать гнездо домена UNIX типа virtual circuit. Послать в серверное гнездо сообщение, содержащее имена всех файлов текущего каталога, созданных за последние 3 дня.

Вариант №19

Сервер. Создать гнездо домена INET типа virtual circuit. Присвоить ему имя. Записать в него информацию об очередях сообщений присутствующих в системе, если они есть. Распечатать ответ клиента.

Клиент. Создать гнездо домена INET типа virtual circuit. Прочитать сообщение из серверного гнезда. Определить очереди сообщений, принадлежащие пользователям, работающим в настоящее время в системе и отправить в серверное гнездо.

Вариант №20

Сервер. Создать гнездо без установления соединения домена INET. При поступлении клиентских сообщений определить файл минимального размера, а также имя владельца этого файла, время последней модификации и права доступа; переслать результат клиенту.

Клиент. Создать гнездо без установления соединения домена INET. Передать в серверное гнездо ряд сообщений, содержащих: а). имена всех файлов текущего каталога представленных на машинном языке; б). суммарный размер этих файлов в байтах. Распечатать ответ сервера.

Список источников

1. Чан Т. Системное программирование на C++ для UNIX. – Киев: Издательская группа BHV, 1999 г.
2. Моли Б. UNIX/LINUX: Теория и практика программирования. - М: КУДИЦ_ОБРАЗ, 2004 г.
3. Роббинс А. LINUX: программирование в примерах. – М: КУДИЦ_ОБРАЗ, 2005 г.
4. Робачевский А.М. Операционная система UNIX. – СПб.: БХВ-Петербург, 2002 г.
5. Истратов А.Ю. Межпроцессное взаимодействие на уровне «клиент-сервер» в ОС UNIX (Учебное пособие), М.: РГУИТиП, 2006 г.

Вопросы для оценки качества освоения дисциплины

1. Понятие «Операционная система».
2. Настройка и поддержка работы сети в ОС UNIX.
3. Расслоение памяти. Регистр перемещений. Прерывания и опрос состояний.
4. Системный вызов sigaction в ОС UNIX.
5. Буферизация. Периферийные устройства. Защита памяти.
6. Особенности программирования на командном языке.
7. Таймер. Каналы ввода/вывода. Захват цикла.
8. Поддержка в ОС UNIX сетевого обмена.
9. Относительная адресация. Режимы работы ЭВМ.
10. Понятие сигнальной маски процесса. Системные вызовы работы с сигнальной маской процесса в ОС UNIX.
11. Виртуальная память. Мультипроцессорная обработка. Прямой доступ к памяти.
12. Программирование на машинном языке. Ассемблеры и макропроцессоры.
13. Библиотечные функции setjmp и longjmp. Сохранение значений при обработке сигнала.

14. Конвейеризация. Иерархия памяти.
15. Прерывания выполнения системных вызовов.
16. Компиляторы.
17. Обработка сигналов в ОС UNIX.
18. Системы управления вводом/выводом. Спулинг.
19. Интерактивный режим работы ОС UNIX.
20. Процедурно-ориентированные и проблемно-ориентированные языки. Интерпретаторы.
21. Начальная загрузка и выход на интерактивный режим в ОС UNIX.
22. Абсолютные и перемещающие загрузчики. Связывающие загрузчики и редакторы связей.
23. Функционирование ОС UNIX.
24. Микропрограммирование. Эмуляция. Горизонтальный и вертикальный микрокод.
25. Реализация операций ввода/вывода в ОС UNIX.
26. Понятие процесса. Состояние процесса.
27. Структурные компоненты системы управления вводом/выводом ОС UNIX.
28. Блок управления процессом. Операции над процессом.
29. Логическая и физическая организация системы управления вводом/выводом ОС UNIX.
30. Понятие прерывания. Схемы прерываний.
31. Физическая и логическая организация файловой системы ОС UNIX.
32. Понятия параллельного процесса, асинхронного процесса, взаимноисключения, семафора, монитора, тупика.
33. Команды файловой системы ОС UNIX.
34. Стратегии управления памятью. Концепции распределения памяти.
35. Типы файлов в файловой системе ОС UNIX. Топология файловой системы.
36. Схемы реализации файлов.
37. Организация памяти при связном распределении. Оверлейные перекрытия.
38. Понятие стандартного ввода/вывода и переназначения стандартного ввода/вывода.
39. Организация памяти на основе мультипрограммирования с переменными разделами.
40. Системные вызовы ввода/вывода.
41. Организация памяти на основе свопинга.
42. Системные вызовы работы с файлами open, creat, close, dup, dup2.
43. Концепции виртуальной памяти.
44. Системные вызовы fork, wait, pipe.
45. Понятия файла, символического набора, организации файлов.
46. Понятие системного вызова. Системные вызовы system и exec1.
47. Средства файловой системы. Организация файловой системы.
48. Информационные виды связи между процессами в ОС UNIX.
49. Блок управления файлом.
50. Структура процесса ОС UNIX. Состояние системы и виды взаимодействия в ОС UNIX.
51. Средства файловой системы. Топология файловой системы.
52. Структура ОС UNIX. Ядро и процессы.
53. Построение локальных вычислительных сетей. Основные аппаратные компоненты.
Основные программные средства.
54. Основные характеристики ОС UNIX.
55. Понятие объединенной локальной сети. Топология ЛВС.
56. Интерпретатор команд shell.
57. Понятия IP-адреса, адреса сети, широковещательного адреса, адреса шлюза, маски сети, адреса сервера имен.
58. Команды файловой системы ОС UNIX.
59. Обмен сообщениями в ОС UNIX.
60. Системные вызовы манипулирования сообщениями в ОС UNIX.
61. Поддержка семафоров в ОС UNIX.

62. Системные вызовы манипулирования семафорами в ОС UNIX.
63. Понятие разделяемой памяти в ОС UNIX.
64. Системные вызовы манипулирования разделяемой областью памяти в ОС UNIX.
65. Обмен сообщениями с помощью разделяемой памяти и семафоров в ОС UNIX.
66. Гнезда в ОС UNIX.
67. Системные вызовы для работы с гнездами в ОС UNIX.
68. Компоненты исполняющей системы ОС WINDOWS.
69. Службы уровня HAL ОС WINDOWS.
70. Функции микроядра ОС WINDOWS.
71. Структура объекта ОС WINDOWS.
72. Подсистемы окружения ОС WINDOWS.
73. Инструменты для управления ресурсами в ОС WINDOWS.
74. Функции диспетчера (менеджера) памяти ОС WINDOWS.
75. Создание процессов и потоков в ОС WINDOWS.
76. Коммуникационные компоненты ОС WINDOWS.
77. Планирование потоков в ОС WINDOWS.
78. Назначение сеансового менеджера ОС WINDOWS.
79. Реализация управления оперативной памятью в ОС WINDOWS.
80. Файловые системы FAT и NTFS.

V. РЕСУРСЫ

1. Основная литература

1. Дейтел Г. Введение в ОС, т.1,2. – М.: Мир, 1987
2. Катцан Г. Операционные системы. – М.: Мир, 1976
3. Девис У. Операционные системы. – М.: Мир, 1980
4. Краковяк С. Основы организации и функционирование ОС ЭВМ. – М.: Мир, 1988
5. Чан Т. Системное программирование на C++ для UNIX. – Киев: Издательская группа BHV, 1999
6. Петерсен Р. LINUX: руководство по ОС, т. 1,2. - Киев: Издательская группа BHV, 1999
7. Истратов А.Ю., Белякова К.Л. Программирование в операционной среде UNIX. – Методические указания для выполнения лабораторных работ. – М.: МИЭМ, 1995
8. Истратов А.Ю., Агеев Д.А., Захарова Н.Ф., Ильиченкова З.В., Мазурина С.М. Основы программирования на C/C++.- М.: Научлиттехиздат, 2002
9. Таненбаум Э. Современные операционные системы. 2-е изд. – СПб.: Питер, 2002
10. Истратов А.Ю. Межпроцессное взаимодействие на уровне «клиент-сервер» в ОС UNIX (Учебное пособие), М.: РГУИТиП, 2006 г.
11. Истратов А.Ю. -Программирование в операционной среде UNIX: обмен информацией между параллельными процессами, организация защиты файлов в файловой системе, обработка прерываний (Учебное пособие), М.: РГУИТиП, 2006 г.

2. Дополнительная литература

1. Грибанов В.П., Дробин С.Г., Медведев В.Д. Операционные системы. – М.: Финансы и статистика, 1990
2. Мэдник С., Донован Дж. Операционные системы. – М.: Мир, 1975
3. Кейлингер П. Элементы ОС. – М.: Мир, 1985

4. Беляков М.И. и др. Инструментальная мобильная ОС ИНМОС. – М.: Финансы и статистика, 1985
5. Браун С. ОС UNIX. – М.: Мир, 1986
6. Готье Р. Руководство по ОС UNIX. – М.: Мир, 1985
7. Дансмур М., Дейвис Г. ОС UNIX и программирование на языке Си. – М.: Финансы и статистика, 1989
8. Кристиан К. Введение в ОС UNIX. – М.: Финансы и статистика, 1995

12. Программное обеспечение

№ п/п	Наименование	Условия доступа
1.	Microsoft Windows 7 Professional RUS Microsoft Windows 10 Microsoft Windows 8.1 Professional RUS	<i>Из внутренней сети университета (договор)</i>
2.	Microsoft Office Professional Plus 2010	<i>Из внутренней сети университета (договор)</i>
3	Операционные системы FREE BSD, LINUX, SOLARIS, MAC и др. UNIX-подобные системы	

13. Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)

№ п/п	Наименование	Условия доступа
	<i>Профессиональные базы данных, информационно-справочные системы</i>	
	<i>Интернет-ресурсы (электронные образовательные ресурсы)</i>	
1.	Открытое образование	URL: https://openedu.ru/

14. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

- ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением.