

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ОПЕРАЦИОННЫЕ СИСТЕМЫ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
01.03.04 «Прикладная математика»
дневной формы обучения*

Часть 1



Могилев 2023

УДК 004.7
ББК 32.973.26
О60

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «28» марта 2023 г., протокол № 9

Составитель ст. преподаватель Е. А. Зайченко

Рецензент Ю. С. Романович

Изложены рекомендации к выполнению лабораторных работ для студентов направления подготовки 01.03.04 «Прикладная математика» дневной формы обучения по дисциплине «Операционные системы». Приведен перечень необходимой литературы.

Учебное издание

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Часть 1

Ответственный за выпуск	В. В. Кутузов
Корректор	Т. А. Рыжикова
Компьютерная верстка	Н. П. Полевнича

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.

Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

Введение.....	4
Общие требования к отчету	5
1 Лабораторная работа № 1. Интерфейс командной строки ОС Windows. Перенаправление потоков ввода/вывода	6
2 Лабораторная работа № 2. Изучение интерпретатора команд ОС Windows	7
3 Лабораторная работа № 3. Работа с процессами и потоками в Windows	9
4 Лабораторная работа № 4. Синхронизация потоков в Windows.....	11
5 Лабораторная работа № 5. Синхронизация процессов и потоков при помощи событий и семафоров в ОС Windows.....	12
6 Лабораторная работа № 6. Изучение архитектуры памяти в ОС Windows	17
7 Лабораторная работа № 7. Сервер сценариев WSH. Основные возможности.....	18
8 Лабораторная работа № 8. Сервер сценариев WSH. Работа с реестром.....	19
9 Лабораторная работа № 9. Сервер сценариев WSH. Работа с устройствами ввода-вывода	21
10 Лабораторная работа № 10. Сервер сценариев WSH. Работа с объектами файловой системы	22
Список литературы.....	24

Введение

Целью дисциплины «Операционные системы» является получение студентами знаний о множестве задач, которые решает операционная система, об особенностях разработки системного программного обеспечения, а также о перспективных направлениях в развитии современных операционных систем.

В результате изучения дисциплины студент узнает:

- способы использования функций операционной системы и администрирования;

- назначение и возможности операционной системы;

- командные средства системного программирования.

Одной из наиболее эффективных форм получения знаний является выполнение лабораторных работ. Защита лабораторной работы состоит из двух частей: практической и теоретической. В практической части студент объясняет алгоритмы и принципы выполненной работы, в теоретической – отвечает на вопросы по теме лабораторной работы. При подготовке к защите студенту рекомендуется ответить на контрольные вопросы.

Общие требования к отчету

Отчет должен содержать стандартные составные части.

1 Титульный лист с указанием следующих реквизитов: название учреждения образования, название закрепленной за дисциплиной кафедры, номер и название лабораторной работы, название дисциплины, вариант, ФИО, группа (выполнившего лабораторную работу), должность, ФИО (проверяющего работу), место и дата составления отчета.

2 Цель работы.

3 Постановка задачи.

4 Выполненное задание согласно варианту: код программы с необходимыми комментариями.

5 В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Обычно программа тестируется на нескольких вариантах входных данных для проверки ее корректности. Каждый рисунок должен располагаться по центру страницы, иметь подпись (Рисунок 1 – Создание подсистемы) и ссылку на него в тексте.

6 Выводы по теме лабораторной работы.

Отчет оформляется шрифтом гарнитуры TimesNewRoman, кегль 14, межстрочный интервал – полуторный, абзацный отступ – 1,25 см.

Страницы должны быть пронумерованы вверху посередине. Титульный лист при нумерации считается, но не нумеруется.

1 Лабораторная работа № 1. Интерфейс командной строки ОС Windows. Перенаправление потоков ввода/вывода

Цель работы: овладеть навыками работы с командной строкой в ОС Windows.

Теоретические сведения

1 Средства работы с командной строкой [2, с. 2–3].

2 Перенаправление стандартных ввода, вывода и ошибок [2, с. 24–28].

Практическое задание

Разработайте два приложения на языке C++ (далее P1 и P2). Варианты приведены в таблице 1.1.

Таблица 1.1 – Варианты для разработки приложений.

Номер варианта	Формула первого приложения	Формула второго приложения
1	$S = \frac{\sqrt{A \cdot B \cdot C}}{\log_2 A}$	$S = \sqrt{S}$
2	$S = \frac{\sqrt{A \cdot B \cdot C}}{\ln A}$	$S = S^2$
3	$S = \sum_i^B (\log_2 A \cdot i)$	$S = \sqrt[3]{S}$
4	$S = \sum_1^B (\sqrt{A} \cdot i)$	$S = S^3$
5	$S = \sum_i^B (\sqrt{A} \cdot C)$	$S = \sqrt[4]{S}$
6	$S = \ln(\sqrt[2]{A} \cdot B^2)$	$S = S^4$
7	$S = \prod_i^A B - i$	$S = \sqrt[5]{S}$
8	$S = \frac{A}{B \cdot \sqrt{C}}$	$S = S^5$
9	$S = \frac{A + B}{\sqrt{C}}$	$S = \sqrt[6]{S}$
10	$S = \frac{A - B}{\sqrt{C}}$	$S = S^6$

Создайте файл source.txt. Он должен содержать такое количество чисел типа XX.XX, какое необходимо передать первому приложению в качестве исходных данных. Введите команду `copy con source.txt`, нажмите Enter, после введите числа через пробел и нажмите Ctrl + Z.

Выполните запуск P1 с вводом исходных данных из подготовленного файла source.txt.

Организуйте конвейер таким образом, чтобы исходные данные для P1 вводились с клавиатуры, а результат выполнения P1 был входными данными для P2.

Организуйте конвейер таким образом, чтобы исходные данные для P1 поступали из файла source.txt, результат выполнения P1 был входными данными для P2.

Организуйте конвейер таким образом, чтобы исходные данные для P1 поступали из файла source.txt, результат выполнения P1 был входными данными для P2, а результат P2 перенаправлялся для записи в файл с расширением .txt, с выполнением дозаписи. Просмотрите результат выполнения, запустив текстовый файл результата из командной строки.

Контрольные вопросы

- 1 Приведите классификацию интерфейсов операционных систем.
- 2 Охарактеризуйте достоинства и недостатки каждого вида интерфейса.
- 3 Для каких целей используется переназначение потоков?
- 4 В каких случаях применяется конвейерная обработка?

2 Лабораторная работа № 2. Изучение интерпретатора команд ОС Windows

Цель работы: овладеть навыками работы с командными файлами в ОС Windows; научиться работать с параметрами командной строки.

Теоретические сведения

- 1 Стандартные операторы и команды сценариев [2, с. 32–36].
- 2 Передача аргументов в сценарии [2, с. 38–39].

Практическое задание

Разработайте собственные пакетные файлы. При создании протокола в задании используйте перенаправление в файл. Общие требования – проверка наличия параметров, обработка ошибок, вывод сообщений об окончании работы командного файла. Варианты приведены в таблице 2.1.

Таблица 2.1 – Варианты заданий

Вариант	Задание
1	Написать пакетный файл, который будет проверять в каталогах, указанных в качестве параметров, наличие файлов с расширением .TMP и удалять их. Если пути поиска не заданы, вывести сообщение. Создать протокол удаляемых файлов
2	Написать пакетный файл, который будет копировать из текущего каталога все файлы с расширением .DRV, кроме одного файла, указанного в качестве второго параметра командной строки, в каталог, указанный первым параметром. Если имя каталога, в который должно производиться копирование, не задано, то вывести сообщение об этом и прервать выполнение файла
3	Написать пакетный файл, который будет проверять наличие в указанном в качестве первого параметра каталоге файлов. Если файлы там есть, копировать их в указанные в качестве остальных параметров каталоги. Если копирование прошло успешно, очистить исходный каталог. Проверять наличие параметров в командной строке, выводить сообщения об ошибках
4	Написать пакетный файл, который будет копировать из текущего каталога все файлы с расширением .txt, кроме одного файла, указанного в качестве первого параметра командной строки, в каталог, указанный вторым параметром. Переписывать только те файлы, которые новее одноименных в каталоге-приемнике. Если произошла ошибка копирования, выдать сообщение и прервать выполнение файла
5	Создать пакетный файл, который выводил бы содержимое каталогов, указанных в качестве параметров командной строки, в файл протокола, находящийся в каталоге, заданном в качестве первого параметра. Проверять наличие параметров в командной строке, выводить сообщения об ошибках
6	Написать пакетный файл, который будет копировать из корневого каталога съемного диска все файлы с расширением .scr в заданные в качестве параметров каталоги. Записывать в файл logcopy.txt дату выполнения операции и имя каждого копируемого файла
7	Написать пакетный файл, который будет осуществлять поиск файла, заданного в качестве первого параметра в каталогах, заданных в качестве остальных параметров. Вести протокол поиска, записывая, в каких каталогах обнаружен файл. Файл протокола сделать скрытым
8	Создать пакетный файл, который выводил бы содержимое каталогов, указанных в качестве параметров командной строки, в файл протокола, находящийся в каталоге, заданном в качестве первого параметра. Проверять наличие параметров в командной строке, выводить сообщения об ошибках
9	Написать пакетный файл, который будет копировать из корневого каталога съемного диска все файлы с расширением .txt в заданные в качестве параметров каталоги. Записывать в файл logcopy.txt имя каждого копируемого файла, дату и время копирования
10	Написать пакетный файл, который будет проверять в каталогах, указанных в качестве параметров, наличие файлов с расширением .TMP и удалять их. Если пути поиска не заданы, вывести сообщение. Создать протокол, содержащий дату и имена удаляемых файлов

Контрольные вопросы

- 1 Перечислите команды, используемые в пакетных файлах.
- 2 Каким образом можно прервать выполнение пакетного файла?

3 Каким образом можно организовать работу с заранее неизвестным количеством параметров командной строки?

4 Как организовать перебор элементов некоторого множества?

3 Лабораторная работа № 3. Работа с процессами и потоками в Windows

Цель работы: овладеть навыками работы с процессами и потоками в Windows.

Теоретические сведения

Процессы и потоки в Windows [4, с. 983–996].

Практическое задание

Определите базовые приоритеты следующих системных процессов: System, smss, csrss, Бездействие (Idle), отразите их значения в отчете.

Напишите программу, запускающую новый процесс (любое приложение). Наблюдайте процесс запуска приложения с помощью утилиты «Системный монитор». Определите количество потоков в запущенном процессе, базовый приоритет процесса, исследуйте изменения текущего приоритета потоков процесса при активизации процесса и при переходе к другим запущенным в системе приложениям.

Разработайте программу согласно варианту.

1 Создать приложение, запускающее три дочерних потока. Дождаться в главном потоке завершения работы одного из запущенных потоков. Каждый запущенный поток выполняет ведение счетчика, предельное значение которого введено в главном потоке. Повысить приоритет 1-го из потоков на 2. Понизить приоритет 2-го из потоков на 1. Вывести значение счетчика каждого из потоков.

2 Создать приложение, запускающее два дочерних потока. Дождаться в главном потоке завершения работы одного из запущенных потоков. Каждый поток выполняет ведение счетчика, предельное значение которого введено в главном потоке. Понизить приоритет одного из потоков на 2, а другого повысить на 2. Приостановить на некоторое время поток, имеющий наименьший приоритет. Вывести значение счетчика каждого из потоков.

3 Создать приложение, запускающее три дочерних потока. Дождаться в главном потоке завершения работы одного из запущенных потоков. Каждый поток выполняет ведение счетчика, предельное значение которого введено в главном потоке. Для первого потока приоритет не изменять, для второго – повысить на 1, для третьего – повысить на 2. Вывести значение счетчика каждого из потоков.

4 Создать приложение, запускающее четыре дочерних потока. Дождаться в главном потоке завершения работы одного из запущенных потоков. Каждый поток выполняет ведение счетчика, предельное значение которого введено в главном потоке. Изменить приоритеты потоков: 1-му потоку повысить на 2, 2-му –

повысить на 1, 3-му – понизить на 1, 4-му – понизить на 2. Вывести значение счетчика каждого из потоков.

5 Создать приложение, запускающее три дочерних потока. Дождаться в главном потоке завершения работы одного из запущенных потоков. Каждый поток выполняет ведение счетчика, предельное значение которого введено в главном потоке. Для 1-го потока приоритет повысить на 1, для 2-го – понизить на 2, для 3-го – не изменять. Вывести значение счетчика каждого из потоков.

6 Создать приложение, запускающее три дочерних потока. Дождаться завершения работы всех запущенных потоков. Каждый поток выполняет ведение счетчика, предельное значение которого введено в главном потоке. Каждый поток выполняет вывод сообщения о том, что он создан. Изменить приоритеты потоков по порядку их создания (первый поток – наибольший приоритет, третий – наименьший). Вывести значение счетчика каждого из потоков.

7 Создать приложение, запускающее пять дочерних потоков. Каждый поток выполняет вывод сообщения об уровне своего приоритета, а затем выводит цифры своего порядкового номера. Повысить приоритет одного из потоков на 2. Понизить приоритет одного из потоков на 1. Принудительно завершить поток, имеющий наименьший приоритет.

8 Создать приложение, запускающее два дочерних потока. Каждый поток выполняет вывод сообщения об уровне своего приоритета, а затем выводит цифры своего порядкового номера. Понизить приоритет одного из потоков на 2 и вывести значение приоритетов потоков. Приостановить на некоторое время поток, имеющий наименьший приоритет.

9 Создать приложение, запускающее три дочерних потока. Каждый поток выполняет вывод сообщения о том, что он создан, а затем выводит цифры своего порядкового номера. Для 1-го потока приоритет не изменять, для 2-го – повысить на 1, для 3-го – повысить на 2. Вывести значение приоритетов потоков в порядке его убывания.

10 Создать приложение, запускающее четыре дочерних потока. Каждый поток выполняет вывод сообщения о том, что он создан, а затем выводит цифры своего порядкового номера. Изменить приоритеты потоков так, чтобы их приоритеты увеличивались по порядку их создания. Вывести сведения о приоритетах потоков и поочередно завершить их работу.

Контрольные вопросы

- 1 Что такое планирование потоков?
- 2 Какие классы приоритетов применяются в ОС Windows?
- 3 В чем отличие базового и текущего приоритетов?
- 4 Перечислите возможные уровни приоритета.
- 5 Что включает в себя создание и завершение процесса?

4 Лабораторная работа № 4. Синхронизация потоков в Windows

Цель работы: овладеть навыками работы для синхронизации потоков в ОС Windows.

Теоретические сведения

- 1 Проблемы многопоточных программ [1, с. 94–107].
- 2 Критические области [4, с. 148–150].

Практическое задание

Задание 1

Написать программу для консольного процесса, который состоит из потока main и двух потоков worker1 и worker2 (по варианту). Синхронизацию потоков выполнить с помощью оператора lock. Протестировать программу без использования lock. Объяснить результаты.

Поток main должен выполнить следующие действия.

- 1 Создать массив случайных целых чисел (как положительных, так и отрицательных), размерность которого вводится с консоли.
- 2 Вывести на консоль исходный массив.
- 3 Создать поток worker1.
- 4 Создать поток worker2.
- 5 Дождаться завершения потоков worker1 и worker2.
- 6 Вывести на консоль массив, полученный после обработки потоком worker1.
- 7 Вывести на консоль результаты работы потока worker2.
- 8 Завершить работу.

Поток worker1 должен выполнить обработку массива (согласно варианту).

Поток worker2 должен найти индексы минимального и максимального элементов обработанного массива.

Задание по вариантам.

- 1 Поток worker1 должен увеличить значения четных элементов массива на 1.
- 2 Поток worker1 должен уменьшить значения четных элементов массива на 5.
- 3 Поток worker1 должен увеличить значения нечетных элементов массива на 10.
- 4 Поток worker1 должен уменьшить значения нечетных элементов массива на 5.
- 5 Поток worker1 должен заменить отрицательные элементы массива на значение модуля элемента.
- 6 Поток worker1 должен заменить элементы, имеющие четный индекс, на 0.
- 7 Поток worker1 должен заменить элементы, имеющие нечетный индекс, на 0.
- 8 Поток worker1 должен заменить четные элементы массива на –5.
- 9 Поток worker1 должен заменить нечетные элементы массива на –5.
- 10 Поток worker1 должен умножить элементы, имеющие четный индекс, на 10.

Задание 2

Написать программу для консольного процесса, который состоит из потока `main` и двух потоков `worker1` и `worker2`.

Поток `main` должен выполнить следующие действия.

Создать два массива случайных целых чисел (как положительных, так и отрицательных), размерность которых вводится с консоли. Создать поток `worker1`. Создать поток `worker2`. Вывести на консоль исходные массивы. Дождаться завершения потоков `worker1` и `worker2`. Вывести на консоль значение, полученное потоком `worker1`. Вывести на консоль результаты работы потока `worker2`. Завершить работу.

Поток `worker1` должен выполнить следующие действия.

Найти сумму положительных элементов первого массива. Найти сумму положительных элементов второго массива. Определить, какое из найденных значений больше.

Поток `worker2` должен выполнить следующие действия.

Найти количество положительных элементов второго массива. Найти количество отрицательных элементов первого массива. Определить, какое из найденных значений больше.

При разработке программы проанализируйте возможность возникновения тупиков.

Контрольные вопросы

- 1 В каком случае необходима синхронизация потоков?
- 2 Каким образом можно обеспечить корректную работу потоков с общим ресурсом?
- 3 Что такое критическая область?
- 4 В чем достоинства и недостатки оператора `lock`?

5 Лабораторная работа № 5. Синхронизация процессов и потоков при помощи событий и семафоров в ОС Windows

Цель работы: научиться использовать средства синхронизации процессов и потоков в ОС Windows.

Теоретические сведения

- 1 Обнаружение взаимоблокировок [1, с. 107–114].
- 2 События. Семафоры. Мьютексы [4, с. 158–171].

Практическое задание

Разработайте приложения для решения задач согласно варианту.

- 1 Написать программы для консольного процесса Boss (Резидент) и консольных процессов Scout (Шпион). Для моделирования передачи сообщений ввести

специальные события, которые обозначают «точку» и «тире», конец сеанса.

Процесс Boss:

- запрашивает у пользователя количество процессов Scout, которые он должен запустить, и запускает заданное количество процессов Scout;
- принимает от каждого процесса Scout сообщение и выводит его на консоль в одной строке. Принимать сообщение может только от одного процесса, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Scout:

- запрашивает с консоли символы «-», «.» (событие «тире», событие «точка») и передает соответствующие события процессу Boss;
- завершает свою работу, когда будет введён символ, обозначающий конец ввода сообщений.

Каждый отдельный процесс открывать в отдельном консольном окне.

2 Написать программы для консольного процесса Boss (Резидент) и консольных процессов Scout (Шпион). Для моделирования передачи сообщений ввести события, которые обозначают «1», «2», конец сеанса для процессов Scout.

Процесс Boss:

- запрашивает у пользователя количество процессов Scout, которые он должен запустить, и запускает заданное количество процессов Scout;
- принимает от каждого процесса Scout сообщение и выводит его на консоль в одной строке. Принимать сообщение может только от двух процессов, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Scout:

- запрашивает с консоли сообщения, состоящее из «1», «2», и передает их (по одному) процессу Boss;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

3 Написать программы для консольного процесса Boss (Резидент) и консольных процессов Scout (Шпион). Для моделирования передачи сообщений ввести специальные события, которые обозначают любые четыре цифры.

Процесс Boss:

- запрашивает у пользователя количество процессов Scout, которые он должен запустить, и запускает заданное количество процессов Scout;
- принимает от каждого процесса Scout сообщение и выводит его на консоль в одной строке. Принимать сообщение может только от трёх процессов, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Scout:

- запрашивает с консоли сообщение, состоящее из цифр, и передает их (по одному) процессу Boss;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

4 Написать программы для консольного процесса Boss и консольных процессов Employee. Для моделирования передачи сообщений ввести специальные события, которые «0», «1», «2», «3», конец сеанса для процессов Employee.

Процесс Boss:

- запрашивает у пользователя количество процессов Employee, которые он должен запустить, и запускает заданное количество процессов Employee;
- принимает от каждого процесса Employee сообщение и выводит его на консоль в одной строке. Принимать сообщение может только от трёх процессов, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Employee:

- запрашивает с консоли сообщения, состоящее из «0», «1», «2», «3», конец сеанса работы и передает (по одному) его процессу Boss;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

5 Написать программы для консольного процесса Administrator и процессов Writer. Для моделирования передачи сообщений ввести специальные события, которые обозначают сообщение «А», сообщение «В» и конец сеанса для процесса Writer.

Одновременно принимать и отправлять сообщения может только три процесса Writer, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Administrator:

- запрашивает у пользователя количество процессов Writer, которые он должен запустить, и запускает заданное количество процессов Writer;
- принимает сообщение от процесса Writer;
- выводит на консоль сообщение;
- принимает от каждого процесса Writer сообщение о завершении сеанса и выводит его на консоль в одной строке.

Процесс Writer:

- запрашивает с консоли сообщения, передает их (по одному) процессу Administrator;
- передает сообщение о завершении сеанса процессу Administrator и завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

6 Написать программы для консольного процесса Boss и консольных процессов Employee. Для моделирования передачи сообщений ввести специальные события, которые обозначают «1», «2», «3» и конец сеанса для процессов Employee.

Процесс Boss:

- запрашивает у пользователя количество процессов Employee, которые он должен запустить, и запускает заданное количество процессов Employee;
- принимает от каждого процесса Employee сообщение и выводит его на консоль в одной строке. Принимать сообщение может только от трёх процессов,

передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Employee:

- запрашивает с консоли сообщения, состоящее из «1», «2», «3», конец сеанса работы и передает (по одному) его процессу Boss;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

7 Написать программы для консольного процесса Administrator и процессов Writer. Для моделирования передачи сообщений ввести специальные события, которые обозначают сообщение «+», сообщение «-», конец сеанса для процесса Writer.

Одновременно принимать и отправлять сообщения может только два процесса Writer, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Administrator:

- запрашивает у пользователя количество процессов Writer, которые он должен запустить и запускает заданное количество процессов Writer;
- принимает сообщение от процесса Writer;
- выводит на консоль сообщение;
- принимает от каждого процесса Writer сообщение о завершении сеанса и выводит его на консоль в одной строке.

Процесс Writer:

- запрашивает с консоли сообщения и передает их по одному процессу Administrator;
- передает сообщение о завершении сеанса процессу Administrator;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

8 Написать программы для консольного процесса Boss и консольных процессов Parent, Child. Для моделирования передачи сообщений ввести события, которые обозначают любые две цифры и конец сеанса для процессов Parent и Child.

Процесс Boss запрашивает у пользователя количество процессов Parent и количество процессов Child, которые он должен запустить, запускает заданное количество процессов Parent, Child и передает каждому не более трёх сообщений, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Parent получает сообщение от процесса Boss и выводит его на консоль.

Процесс Child получает сообщение от процесса Boss и выводит его на консоль.

Каждый отдельный процесс открывать в отдельном консольном окне.

9 Написать программы для консольного процесса Administrator и консольных процессов Reader и Writer. Для моделирования передачи сообщений ввести специальные события, которые обозначают сообщение «А», сообщение «В», конец сеанса для процессов Reader и Writer.

Одновременно принимать и отправлять сообщения могут только два процесса Writer и два процесса Reader, передача остальных сообщений от других процессов должна блокироваться с помощью семафоров.

Процесс Administrator:

- запрашивает у пользователя количество процессов и запускает заданное количество процессов Reader и Writer;
- принимает от каждого процесса Writer сообщение и выводит на консоль, затем отправляет его процессу Reader;
- принимает от каждого процесса Reader и Writer сообщение о завершении сеанса и выводит его на консоль в одной строке.

Процесс Writer:

- запрашивает с консоли сообщения, состоящее из «А», «В», передает их по одному процессу Administrator;
- передает сообщение о завершении сеанса процессу Administrator;
- завершает свою работу.

Процесс Reader:

- принимает сообщение от процесса Administrator;
- выводит на консоль сообщение;
- передает сообщение о завершении сеанса процессу Administrator;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

10 Написать программы для консольного процесса Boss и консольных процессов Parent, Child. Для моделирования передачи сообщений ввести специальные события, которые обозначают «С», «D», конец сеанса для процессов Parent и Child.

Процесс Boss:

- запрашивает у пользователя количество процессов Parent и количество процессов Child, которые он должен запустить и запускает заданное количество процессов Parent, Child;
- принимает от каждого процесса Parent, Child сообщение и выводит на консоль в одной строке не более четырех сообщений, передача остальных сообщений от других процессов должна блокироваться с помощью семафора.

Процесс Parent:

- запрашивает с консоли сообщения, состоящее из «С», передает их по одному процессу Boss;
- завершает свою работу.

Процесс Child:

- запрашивает с консоли сообщения, состоящее из «D», передает их по одному процессу Boss;
- завершает свою работу.

Каждый отдельный процесс открывать в отдельном консольном окне.

Контрольные вопросы

- 1 Для каких целей используются объекты-события, в чем отличие «автоматических» и «ручных» событий?
- 2 В чем отличие событий и семафоров?
- 3 Перечислите методы, позволяющие выполнить ожидание объектов синхронизации.

6 Лабораторная работа № 6. Изучение архитектуры памяти в ОС Windows

Теоретические сведения

Цель работы: получить представление об алгоритмах и методах управления памятью в ОС Windows.

Теоретические сведения

- 1 Архитектура памяти в Windows [1, с. 155–158].
- 2 Управление памятью [4, с. 1003–1009].

Практическое задание

1 С помощью Диспетчера задач определите текущие значения всех статистических параметров памяти. Запустите до 10 приложений и определите узкое место в системе (ОЗУ или ЦП) путем анализа графиков Хронология использования памяти и Хронология загрузки ЦП. Запишите новые значения статистических параметров памяти: *используемая память, доступная, выделено, кешировано, выгружаемый* и *невыгружаемый* пулы. Закройте открытые приложения и запишите новые значения статистических параметров памяти, сделайте выводы. Каково максимальное значение используемой памяти? Сравните с прежним его значением и сделайте выводы.

2 Запустите приложения MS Word, MS Excel, Visual Studio. С помощью Диспетчера задач определите объемы памяти, используемые процессами. Определите, как изменяются эти параметры при изменении активности приложений.

3 Изучите справочную информацию о параметрах запуска утилиты TaskList. Получите с помощью утилиты информацию об используемой оперативной памяти каждым процессом системы. Запустите приложения MS Word и MS Excel. Получите с помощью утилиты TaskList информацию о PID их образов и список всех модулей, загруженных в оперативную память и используемых этими процессами. Определите, есть ли совместно используемые модули.

4 Запустите Монитор ресурсов. Определите полный объем физической памяти в компьютере, общий объем виртуальной памяти, доступной (свободной) в данный момент времени виртуальной памяти.

Просмотрите сведения об использовании физической памяти аппаратными компонентами компьютера; определите диапазон адресов памяти, используемый

несколькими из них.

Запустите несколько приложений и определите используемый ими объем ОП. Определите, какой процесс использует наибольшее количество памяти совместно с другими процессами (разделяемый ресурс).

Наблюдая показатель «Ошибок отсутствия страницы в памяти/сек», определите, в каком случае этот показатель увеличивается. Объясните полученные результаты.

5 Запустите Системный монитор. Создайте журнал счетчиков, позволяющие оптимизировать виртуальную память (память / доступно байт, память / обмен страниц в сек, файл подкачки / % использования) и проведите наблюдение за ситуациями, порождающими недостаток памяти. Запустите журнал счетчиков и некоторое время понаблюдайте за системой. Выполните анализ полученных результатов и дайте рекомендации по улучшению конфигурации ПК.

6 Определите, используется ли операционной системой файл подкачки, его расположение, минимальный и рекомендуемый объем. Найдите на диске (дисках) файл подкачки и определите его размер.

Контрольные вопросы

- 1 Чем определяется размер виртуального адресного пространства процесса?
- 2 На какие части ОС Windows распределяет адресное пространство процесса?
- 3 Как происходит выделение VMM памяти для процесса?
- 4 Как происходит запуск на исполнение EXE-модуля?
- 5 Для каких целей используются файл подкачки и файл гибернации?
- 6 Что такое «ошибки страниц»?

7 Лабораторная работа № 7. Сервер сценариев WSH. Основные возможности

Цель работы: научиться работать с сервером сценариев ОС Windows, изучить возможности сервера сценариев, режимы выполнения сценариев.

Теоретические сведения

- 1 Сервер сценариев WSH. Языки сценариев VBScript и JScript [3, с. 52–56].
- 2 Собственная объектная модель WSH [3, с. 62–76].

Практическое задание

Создайте файлы сценариев, которые выполняют следующие операции.

- 1 Выводят сведения о системе (таблица 7.1).
- 2 Выводят информацию о локальном компьютере, на котором работает студент (имя пользователя и имя компьютера).
- 3 Выводят сведения о сетевых принтерах и их количество. Если сетевых принтеров несколько, запросить у пользователя, какой из них следует сделать

используемым по умолчанию.

4 Выполняют попытку подключения сетевого диска с заданным пользователем именем и параметрами. Если подключение прошло успешно, перед завершением работы сценария предложить отключить сетевой диск.

Таблица 7.1 – Варианты сценариев

Вариант	Сведения о системе
1	Операционная система, директория Windows
2	Расширенные данные о процессоре, поколение процессора
3	Заданные системные пути, директория Windows
4	Тип процессора, операционная система
5	Количество процессоров на данном компьютере, заданные системные пути
6	Диск, на котором находится директория с операционной системой, каталог по умолчанию для пользователей
7	Первый локальный диск, поколение процессора
8	Путь к файлу командной строки, папка для временных файлов
9	Тип процессора, заданные системные пути
10	Заданные системные пути, поколение процессора

Контрольные вопросы

- 1 Назначение сервера сценариев.
- 2 С помощью каких методов объекта WshShell можно получить информацию об аппаратном обеспечении компьютера?
- 3 Как устроены множества для работы с сетевыми дисками и принтерами?

8 Лабораторная работа № 8. Сервер сценариев WSH. Работа с реестром

Цель работы: изучить структуру реестра Windows, способы резервирования данных реестра, возможности сервера сценариев по работе с реестром.

Теоретические сведения

Сервер сценариев WSH. Работа с системным реестром [3, с. 92–132].

Практическое задание

Изучите структуру реестра с помощью утилиты Regedit. Найдите раздел, содержащий сведения, соответствующие варианту (таблица 8.1). Выполните экспорт данного раздела.

Создайте сценарий WSH, выполняющий поиск (чтение) заданного пользователем расширения файла (например, «.7z»), в реестре. Если соответствующий

раздел в реестре имеется, вывести его значение. Используйте обработку исключительных ситуаций.

Ознакомьтесь с командами утилиты reg. Создайте командный файл, осуществляющий запуск утилиты Reg и выполнение резервного копирования части реестра, содержащей следующие сведения (таблица 8.1). Выводите информацию о том, какие сведения копируются, в какой файл.

Таблица 8.1 – Варианты сценариев

Вариант	Сведения в реестре
1	Настройки рабочего стола пользователя, на данный момент зарегистрированного в системе
2	Сведения о настройках для профиля пользователя по умолчанию
3	Данные об аппаратных средствах компьютера
4	Сведения о профиле пользователя
5	Данные о процессоре
6	Данные о программном обеспечении, установленном на компьютере
7	Данные о системной памяти компьютера
8	Данные об программных средствах компьютера
9	Сведения о профилях пользователей
10	Данные о принтере, установленном для текущего пользователя

Контрольные вопросы

- 1 Какую структуру имеет реестр?
- 2 Как используют реестр программы установки и удаления программного обеспечения?
- 3 Каким образом реестр используется для взаимодействия ОС с внешними устройствами?
- 4 Что представляют собой аппаратные и пользовательские профили?
- 5 Перечислите основные разделы реестра и их назначение.
- 6 В каких файлах хранятся данные реестра? В чем отличие файлов с расширением .log и .sav?
- 7 Перечислите основные возможности редактора реестра.
- 8 Для чего используется утилита Reg?
- 9 Перечислите основные возможности сервера сценариев WSH для работы с реестром.

9 Лабораторная работа № 9. Сервер сценариев WSH. Работа с устройствами ввода-вывода

Цель работы: изучить структуру реестра Windows, способы резервирования данных реестра, возможности сервера сценариев по работе с реестром.

Теоретические сведения

- 1 Сервер сценариев WSH. Инвентаризация оборудования [3].
- 2 Ввод-вывод [4, с. 304–307, 315–334].

Практическое задание

1 Создайте сценарий, выполняющий запись в файл сведений о материнской плате: производитель, тип первичной шины системной платы, тип вторичной шины системной платы, тип шины (1 - ISA, 5 - PCI Bus, 15 - PNP Bus и т. д.). Используйте классы Win32_BaseBoard, Win32_MotherboardDevice, Win32_OnBoardDevice, Win32_Bus, Win32_SystemSlot.

2 Создайте сценарий, выполняющий запись в файл сведений о количестве процессоров и скорости процессора, о размерах кеша 2-го уровня. Используйте классы Win32_Processor, Win32_CacheMemory.

3 Создайте сценарий, выполняющий запись в файл сведений о количестве свободной физической памяти, о диапазоне доступных адресов. Используйте классы Win32_PhysicalMemory, Win32_MemoryDevice, Win32_DMACHannel.

4 Создайте сценарий, выполняющий запись в файл сведений обо всех портах ввода-вывода (I/O ports), найденных на компьютере. Используйте классы Win32_PortResource, Win32_ParallelPort, Win32_SerialPort.

5 Создайте сценарий, выполняющий запись в файл сведений о разрешении экрана, наименовании клавиатуры и количестве функциональных клавиш. Используйте классы Клавиатура (Win32_Keyboard), Монитор (Win32_DesktopMonitor).

6 Создайте сценарий, выполняющий запись в файл сведений о наименовании, производителе и количестве кнопок мыши, характеристиках шины системной платы. Используйте классы Мышь (Win32_PointingDevice), Шина (Win32_Bus).

7 Создайте сценарий, выполняющий запись в файл сведений о идентификаторах и производителях устройств Plug-and-Play, наименование и производители аудиоустройств. Используйте классы Аудио (Win32_SoundDevice), устройства Plug-and-Play (Win32_PnPEntity).

8 Создайте сценарий, выполняющий запись в файл сведений о качестве цветопередачи (количество бит на пиксель), частоте обновления экрана (Гц), описание видеопроцессора. Используйте классы Видео (Win32_VideoController), Класс CIM_VideoControllerResolution.

9 Создайте сценарий, выполняющий запись в файл сведений о файловых системах логических дисков, о IP-адресе компьютера. Используйте классы Win32_LogicalDisk, Win32_NetworkAdapterConfiguration.

10 Создайте сценарий, выполняющий запись в файл сведений о том, имеются ли на компьютере CD-ROM, его марка, об устройствах, подключенных к USB-портам. Используйте классы Диски (Win32_DiskDrive), CD-ROM (Win32_CDRomDrive), USB-порт (Win32_USBHub).

Контрольные вопросы

- 1 На какие категории делятся устройства ввода-вывода?
- 2 Перечислите основные функции контроллера.
- 3 Какие операции выполняет драйвер в подсистеме ввода-вывода?
- 4 Что такое прерывание?
- 5 Какими способами ОС реагирует на прерывания?
- 6 Перечислите основные задачи программного слоя подсистемы ввода-вывода.
- 7 Перечислите основные функции драйвера.
- 8 Объясните принципы технологии PnP.

10 Лабораторная работа № 10. Сервер сценариев WSH. Работа с объектами файловой системы

Цель работы: изучить объекты файловой системы, основные команды для работы с файлами и каталогами.

Теоретические сведения

- 1 Сервер сценариев WSH. Работа с файловой системой [3, с. 222–241].
- 2 Собственная объектная модель WSH [3, с. 62–76].

Практическое задание

Разработайте сценарий, выполняющий действия, согласно варианту.

1 Проверить наличие и доступность хотя бы одного съемного диска (в противном случае вывести сообщение об ошибке). Создать на нем папку и записать в эту папку текстовый файл протокола. В этот файл записать сведения о папке пользователя на системном диске: дату последнего изменения, размер, количество папок. После записи сведений установить для файла протокола атрибут «только чтение».

2 Пользователь задает путь к папке. Проверить наличие заданного пути. Создать новую папку, установив для нее заданные пользователем атрибуты, и скопировать в нее файлы из корневого каталога диска, заданного пользователем. Записать сведения о скопированных файлах в файл протокола.

3 Вывести вопрос, какую папку создать и на каком диске. Проверить, есть ли такой диск. В папке создать файл Inform.txt и записать в него информацию о заданном пользователем диске: тип файловой системы и количество доступного для пользователя места. Если такой файл уже имеется, дописать в него информацию и присвоить ему атрибут «Скрытый».

4 Проверить, были ли обращения к заданной папке со времени, введенного пользователем. Вывести список и общий размер файлов с атрибутом «Только для чтения», имеющихся в папке. Создать файл и записать в него информацию о заданном пользователем диске: метку диска, серийный номер и общий размер.

5 Пользователь задает имя папки на сетевом диске. Проверить, доступен ли указанный диск и существует ли папка. Если со дня создания папки к ней не было обращений больше месяца, предложить удалить ее файлы, находящиеся в ней (по одному), выводя размер и дату создания каждого файла. Если в папке есть файлы с расширением .txt, выводить их первые три строки.

6 Определить, изменялась ли заданная с помощью полного пути папка за последнюю неделю. Если нет, создать файл Info.txt, записав туда тип диска, на котором находится эта папка, его метку, дату последнего обращения и изменения этой папки. Файлу присвоить атрибут «Только чтение».

7 Пользователь задает путь к папке (возможно на сетевом диске). Проверить ее наличие и доступность. Определить количество и объем файлов, входящих в папку, кроме системных и скрытых, и записать эту информацию в файл на съемном диске (если он доступен).

8 Проверить, имеется ли на компьютере доступный съемный диск. Если да, создать файл «протокол. txt» и записать в него информацию о жестких дисках – общий размер, занятое пространство, свободное пространство, тип файловой системы. Установить на этот файл атрибут «Скрытый». Если файл с таким именем уже есть, перезаписать его.

9 Определить размер файлов с атрибутом «Скрытый», входящих в указанную пользователем папку. Если в ней есть вложенные папки, их размер не учитывать. Эту информацию вывести в файл, который записать в корневой каталог заданного пользователем диска. Размер файлов выводить в соответствующих единицах.

10 Написать сценарий, который будет создавать отчет об использовании дискового пространства на всех доступных дисках. В файл отчета записывать дату проверки, букву диска, метку тома, общий объем, количество свободного и занятого пространства. Записать сообщение, если в корневом каталоге диска обнаружены каталоги, размер которых превышает заданную величину, например, 500 Мегабайт.

Контрольные вопросы

- 1 Что входит в состав файловой системы?
- 2 В чем отличие полного и относительного имени файла?
- 3 Все ли атрибуты файла могут быть изменены пользователем?

4 В чем отличие объекта, полученного с помощью метода OpenFileDialog и объекта, полученного с помощью метода GetFile?

Список литературы

1 **Назаров, С. В.** Операционные системы. Практикум: учебное пособие / С. В. Назаров, Л. П. Гудыно, А. А. Кириченко. – Москва : КНОРУС, 2012. – 376 с.

2 **Станек, У. Р.** Командная строка Microsoft Windows: справочник администратора / У. Р. Станек. – Москва: Русская редакция, 2014. – 480 с.

3 **Попов, А. В.** Командная строка и сценарии Windows / А. В. Попов. – Москва: НОУ Интуит, 2016. – 320 с.

4 **Таненбаум, Э.** Современные операционные системы / Э. Таненбаум, Х. Бос. – Санкт-Петербург: Питер, 2015. – 1120 с.