Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт
расчетно-графического задания (работы)

по дисциплине «Операционные системы», 4 семестр

1. Методика оценки

Выполнение расчетно-графического задания (работы) (далее - РГЗ(Р)) является формой текущей аттестации (контроля) по дисциплине, предусмотренной учебным планом.

Цель РГЗ(Р): студенты должны освоить и научиться программной реализации принципов, методов и алгоритмов, используемыфх в современных ОС

Обязательным элементом РГЗ(Р) являются решение задач, программирование, моделирование, получение и анализ данных измерений и построение графиков

Номер задания выбирается рандомно.

РГЗ(Р) выполняется индивидуально

Количество заданий достаточно для обеспечения, каждого обучающегося индивидуальным заданием РГЗ(Р).

Замена задания РГЗ(Р) осуществляется по согласованию с преподавателем из числа резервных (не занятых) заданий.

Преподаватель осуществляет руководство по выполнению задания, оказывает консультационную помощь и принимает отчет по РГЗ(Р).

По результатам выполнения РГЗ(Р) выполняется отчет, который состоит из следующих частей:

1. Титульный лист.
2. Теоретическая часть, представляющая собой описание структур данных и алгоритмов.
3. Практическая часть, описание структуры программного кода, процесса моделирования, полученных результатов измерений и их анализ.
4. Выводы.
5. Список использованной литературы, интернет-источников и программных средств.

Требования к оформлению:

Объем РГЗ(З) до 15 страниц машинописного текста формата А4. Шрифт Times New Roman, 12. Формулы набираются в редакторе Math Type. Размещение сканированных формул не допускается. Нумерация страниц сквозная, в нижней части листа по центру арабскими цифрами. Работа должна быть отредактирована, не содержать орфографических, синтаксических и стилевых ошибок.

Отчет в установленные сроки сдается на кафедру для проверки. Преподаватель оценивает качество работы, отмечает положительные стороны и недостатки работы и определяет, допускается ли она к защите. При необходимости преподаватель возвращает РГЗ(Р) студенту для доработки и устанавливает сроки повторного предоставления для проверки. До защиты работы студентом должны быть сделаны необходимые исправления и дополнения по всем замечаниям преподавателя.

При положительном результате оценивания РГЗ(Р) студент её распечатывает, передает на кафедру и защищает до сессии в назначенное преподавателем время.

Защита РГЗ(Р) состоит в индивидуальном устном собеседовании студента с преподавателем. В процессе защиты выявляется уровень знаний студента, степень его самостоятельности при выполнении работы. По результатам защиты студенту выставляется оценка в соответствии с критериями, приведенными в п. 2 настоящего Паспорта.

2. Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций

Общие правила выставления оценок текущей аттестации определяются балльно-рейтинговой системой, установленной локальным актом НГТУ.

РГЗ(Р) считается выполненной **на продвинутом** уровне, если все необходимые расчеты произведены самостоятельно и без ошибок; все разделы РГЗ(Р) выполнены правильно и в полном объеме; работа оформлена в соответствии с требованиями; сдана преподавателю в указанные сроки и не возвращалась для доработки; даны полные и развернутые выводы и рекомендации; на защите студентом даны уверенные и аргументированные ответы. Совокупность запланированных результатов и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Закрепленные за РГЗ(Р) компетенции сформированы на продвинутом уровне. Оценка составляет *от 21 до 30 баллов.*

РГЗ(Р) считается выполненной **на базовом** уровне, если все необходимые расчеты произведены самостоятельно и без существенных ошибок; все разделы РГЗ(Р) выполнены правильно, но есть замечания к полноте предоставления информации; работа оформлена в соответствии с требованиями; сдана преподавателю в указанные сроки и однократно возвращалась студенту для незначительной доработки; в заключении даны выводы и рекомендации; на защите студентом допущены непринципиальные ошибки. Совокупность запланированных результатов и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Закрепленные за РГЗ(Р) компетенции сформированы на базовом уровне. Оценка составляет *от 15 до 20 баллов.*

РГЗ(Р) считается выполненной **на пороговом** уровне, если все необходимые расчеты произведены самостоятельно, но с ошибками, часть из которых носит принципиальный характер; есть замечания к полноте предоставления информации; работа оформлена в соответствии с требованиями; сдана преподавателю, но неоднократно возвращалась студенту для доработки; в заключении даны краткие выводы; защита РГЗ(Р) вызывает у студента серьезные затруднения. Совокупность запланированных результатов и соотнесенных с ними индикаторов достижения компетенций содержит существенные пробелы. Закрепленные за РГЗ(Р) компетенции сформированы на пороговом уровне. Оценка составляет *от 10 до 14 баллов.*

РГЗ(Р) считается **не выполненной** (ниже порогового уровня), если расчеты произведены с серьезными ошибками; есть замечания к полноте предоставления информации и оформлению; РГЗ(Р) была сдана преподавателю, но неоднократно возвращалась студенту для доработки, что не привело к улучшению ее качества; РГЗ(Р) не допущена до защиты, что свидетельствует о неудовлетворительном уровне достигнутых студентом результатов. Совокупность запланированных результатов и соотнесенных с ними индикаторов достижения компетенций содержит множественные существенные пробелы. Закрепленные за РГЗ(Р) компетенции не сформированы. Оценка составляет менее 10 баллов.

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

РГЗ(Р) как форма текущей аттестации (контроля) по дисциплине считается успешно выполненной, если сумма полученных баллов по всем его заданиям составляет от 9 до 18 баллов включительно.

4. Примерный перечень и содержание тем РГЗ(Р) «Моделирование структур данных и алгоритмов операционных систем».

4.1. Моделирование файловой системы

Программа моделирует в двоичном файле файловую системы (работы может быть выполнена группой студентов, при разработке каждым из них своего уровня).

Физический уровень. Моделируется чтение/запись двоичного файла блоками по 512 байт (секторами). Реализуется:

1. первичное планирование носителя (структура носителя), размещение основных компонентов
2. управление свободной памятью (FAT-таблица, список секторов с номерами свободных блоков (UNIX), битовая карта, таблица extend-блоков переменной длины).
3. управление размещением файла (FAT-таблица, многоуровневый массив указателей (UNIX), таблица extend-блоков переменной длины)

Базовый уровень. Индексный файл. Набор операций с файлом по его индексу (создать пустой файл, читать сектор по логическому номеру, добавить сектор, обновить сектор, читать файл целиком в динамическую память, запись всего файла из памяти).

Логический уровень. Создание системы каталогов. Преобразование – путь – индекс, создание и уничтожение каталога.

Символьный и пользовательский уровни. Команды форматирования, создания, уничтожения и просмотра каталога и файла, экспорт и импорт файла, информация о внутренней структуре файловой системы.

4.2. Моделирование структур данных операционных систем

1. Каждый процесс имеет описатель, содержащий двухуровневый динамический массив указателей на связанные с ним объекты ОС (файлы, семафоры и т.п.). Объект идентифицируется 16-разрядным handle. Старший байт handle представляет собой смещение в массиве указателей верхнего уровня, младший байт – нижнего. Свободные handle обозначены NULL-указателями. Функция связывания заданного объекта с процессом с выделением ему handle и функция освобождения объекта по заданному handle. Каждый объект имеет счетчик связей. Если объект ОС не имеет больше связей с процессами, то он уничтожается. main моделирует фиксированную последовательность установления и освобождения связей между объектами и процессами (и производит трассировку текущего состояния системы).
2. Каждый процесс (поток) может ожидать событий от нескольких объектов одновременно. С каждым объектов может быть связано несколько ожидающих процессов. Смоделировать структуру связей «процессы-объекты» с использованием в каждом процессе динамического массива указателей на объекты

и в каждом объекте – динамического массива указателей на ожидающие процессы. Реализовать действия: установить ожидание процессом заданного события, смоделировать наступление события в объекте с деблокированием связанных процессов, для которых это событие – единственное (последнее).

3. Задание аналогичное предыдущему, каждый процесс содержит заголовок списка объектов, на которых он ожидает событие, каждый объект содержит список ожидающих процессов. Факт «процесс ожидает событие» реализован элементом, имеющим 4 указателя и включенным одновременно в список процесса и в список объекта.
4. Описатель потока (процесса) представляет собой элемент односвязного списка, содержащего базовый и максимальный приоритеты (в диапазоне 0-255). Диспетчер поддерживает 256 приоритетных очередей (массив указателей на списки процессов). Модель поддерживает также список блокированных (ожидающих) процессов. Каждый процесс имеет две случайные характеристики: количество квантов работы и количество квантов блокировки. Диспетчер моделирует поведение процессов и собственный алгоритм. Если время блокировки процесса истекло, то диспетчер извлекает описатель процесса из очереди ожидающих и помещает его в конец очереди в соответствии с его максимальным приоритетом. Затем ищет очередь с максимальным приоритетом, выбирает из нее первый описатель и «выполняет процесс», после чего переносит его в очередь с более низким приоритетом, если его текущий приоритет выше базового. Если количество квантов работы процесса истекло, он переносится в список блокированных, и для него устанавливаются новые случайные значения квантов.
5. Задание аналогичное предыдущему, но диспетчер поддерживает единственную общую очередь всех процессов - двусвязный список, из которого извлекается первый процесс с максимальным приоритетом.
6. Драйвер устройства содержит таблицу функций (в том числе, инициализация ввода-вывода и обработки прерывания ввода-вывода) и заголовок очереди запросов ввода-вывода. Функция инициализации ввода-вывода устанавливает случайное значение времени работы устройства. main моделирует появление операции ввода-вывода и формирует случайный интервал времени до следующей операции. Диспетчер ввода-вывода создает запрос ввода-вывода и помещает его в очередь драйвера, если она была пуста, вызывает функцию инициализации ввода-вывода в драйвере. Обработчик прерывания в драйвере вызывает функцию завершения ввода-вывода в диспетчере. Диспетчер удаляет запрос из очереди, и, если она не пуста, вызывает функцию инициализации для следующего запроса.
7. Модель древовидной структуры именованных объектов в памяти. Базовый класс объектов содержит имя объекта и ссылки в списке. Производный класс объектов – каталогов содержит заголовок списка включенных в него объектов, а также указатель на каталог-предок. Имеются также производные классы объектов – процессы, файлы, семафоры и т.д.. Разработать редактор дерева объектов и функцию поиска объекта по спецификации <имя>/<имя>/<имя>.
8. Задание аналогичное предыдущему, но объект-каталог использует динамический массив указателей на объекты, включенные в каталог.

4.3. Моделирование процессов и обработки прерываний

При решении задач, связанных с обработкой прерывания используется моделирование физически параллельных процессов в виде потоков на Java.

1. Очередь отложенных прерываний реализована в виде односвязного списка. Заголовок списка представляет собой два указателя на первый и последний элементы. Элемент списка содержит указатель на «отложенную» функцию, которую следует выполнить по завершении прерывания на низком приоритете (при разрешенном прерываний). Функция генерации отложенного прерывания создает элемент списка, помещает в него указатель и ставит его в очередь. Если очередь была пуста, то входит в режим обслуживания очереди прерываний на низком приоритете (при разрешенном прерывании). Поведение системы рассмотреть на основе источника отложенных прерываний – таймера, а в сами функции отложенного прерывания внести случайную программную задержку.
2. Ввод по прерыванию с буферизацией. Имеется список свободных буферов (элементов списка). По прерыванию от таймера генерируется очередное случайное число и записывается в массив (сам массив находится в элементе списка). При заполнении массива буфер помещается в очередь накопленных данных. main производит периодический опрос очереди (используя случайную задержку – функция delay) , выводит полученные данные и освобождает буфер.
3. Модель процессов – функции, запускаемые событиями. Система поддерживает список: элемент списка содержит указатель на запускаемую функцию, ее фактические параметры и значение задержки перед запуском. Опрос клавиатуры и мыши приводят к планированию фиксированных функций. С помощью модели реализовать игровую интерактивную программу. Функция может также рассылать сообщение всем стоящим в очереди функциям.
4. Модель процессов – классы, обрабатывающие сообщения. Базовый класс программных объектов, обрабатывающих сообщения имеет статический список, в который они включаются конструктором и исключаются деструктором, а также виртуальную функцию обслуживания сообщений и функцию – послать сообщение. Класс сообщений содержит указатель на элемент в списке сообщений и ряд параметров, в том числе – код класс – источник сообщения. Класс «программа» поддерживает очередь сообщений и имеет доступ к очереди объектов. Метод run опрашивает таймер, клавиатуру и мышь и генерирует сообщения от них. Полученные сообщения «пропускает» через все программные объекты. Объект, принимающий сообщение, сообщает об этом, после чего обработка сообщения прекращается. С помощью модели реализовать игровую интерактивную программу.