

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гнатюк Максим Александрович
Должность: Первый проректор
Дата подписания: 11.07.2022 09:51:21
Уникальный программный ключ:
8873f497f100e798ae8c92c0d38e105c818d5410

Приложение
к рабочей программе дисциплины

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Системное программирование

(наименование дисциплины(модуля))

Направление подготовки / специальность

09.03.02 Информационные системы и технологии

(код и наименование)

Направленность (профиль)/специализация

Информационные системы и и технологии на транспорте

(наименование)

Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации: зачет в 4 семестре.

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-1 Способен создавать инструментальные средства программирования	ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня
	ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся знает: основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.	Вопросы (1-10)
	Обучающийся умеет: использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач	Вопросы (8-19)
	Обучающийся владеет: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ	Вопросы (14-25)
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся знает: способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства	Вопросы (20-27)
	Обучающийся умеет: отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.	Вопросы (25-38)
	Обучающийся владеет: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.	Вопросы (38-48)

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) ответ на билет, состоящий из теоретических вопросов и практических заданий;
- 2) выполнение заданий в ЭИОС СамГУПС.

2. Типовые¹ контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся знает: основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.
<i>Примеры вопросов/заданий</i> <ol style="list-style-type: none">1. Что относится к ресурсам ЭВМ.2. Организация памяти ЭВМ.3. Оперативная память как ресурс.4. Система регистров микропроцессора и их использование в качестве ресурсов.5. Система адресации в современных компьютерах.6. Процессы.7. Потoki.8. Алгоритмы работы с памятью.9. Алгоритмы планирования потоков.10. Диспетчеризация.11. Многопоточные приложения.	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся знает: способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства
<i>Примеры вопросов/заданий</i> <ol style="list-style-type: none">1. Понятие о библиотеках функций.2. Статические библиотеки.3. Динамические библиотеки.4. Способы компиляции, сборки и использования статических библиотек.5. Способы компиляции, сборки и использования динамических библиотек.6. Сборка многофайловых программ.7. Основы кроссплатформенной разработки приложений.8. Переносимость приложений на уровне исходных кодов.9. Использование для разработки приложений кроссплатформенных библиотек.10. Способы условной компиляции. Использование препроцессора11. Стандартная библиотека функций.12. Использование стандартной библиотеки для программирования многопоточных приложений.	

¹ Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся умеет: использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач
<p><i>Примеры заданий</i></p> <p>1. Системная программа – это:</p> <p>a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации</p> <p>b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.</p> <p>c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.</p> <p>d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.</p> <p>2. Прикладная программа – это</p> <p>a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации.</p> <p>b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.</p> <p>c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.</p> <p>d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.</p> <p>3. Программный модуль — это</p> <p>a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации.</p> <p>b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.</p> <p>c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.</p> <p>d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.</p> <p>4. Объектный модуль — это</p> <p>a. программный модуль, представленный в форме, пригодной для загрузки в оперативную память для выполнения.</p> <p>b. программный модуль, получаемый в результате трансляции исходного модуля.</p> <p>c. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.</p> <p>d. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.</p> <p>5. Программа обслуживания (утилита) — это</p> <p>a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации.</p> <p>b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.</p>	

<p>с. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.</p> <p>d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.</p>	
<p>ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня</p>	<p>Обучающийся владеет: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ</p>
<ol style="list-style-type: none"> 1. Методы и алгоритмы управления ресурсами вычислительной системы. 2. Использование многопоточности в приложениях. 3. Управление процессами. 4. Управление потоками. 5. Компиляция многопоточных приложений. 6. Использование командной строки для создания многопоточных приложений. <p style="text-align: center;">Задание</p> <ol style="list-style-type: none"> 1. Выполнить проектирование многопоточного приложения используя нотацию UML. 2. Произвести компиляцию приложения, используя командную строку. 3. При необходимости исправить ошибки и повторить компиляцию. 4. Выполнить запуск программы и убедиться в ее работоспособности. 	
<p>ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня</p>	<p>Обучающийся умеет: отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.</p>
<ol style="list-style-type: none"> 1. Особенности работы с памятью программ. Операторы выделения блоков памяти и удаления выделенных блоков памяти. 2. Особенности работы с процессами. Создание процесса. Удаление процесса. 3. Особенности работы с потоками. Создание потоков и управление ими. 4. Использование средств управления потоками. 5. Семафоры и мьютексы. 6. Использование API функций операционной системы Windows. 7. Использование системных вызовов операционной системы Linux. 8. Особенности тестирования системных программ. 9. Особенности компиляции программ на уровне ядра операционной системы. 	
<p>ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня</p>	<p>Обучающийся владеет: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.</p>
<ol style="list-style-type: none"> 1. Использование стандартной библиотеки для создания кроссплатформенных многопоточных приложений. 2. Ключевые режимы компилятора для создания системных приложений. 3. Особенности программирования программных библиотек в операционной среде Windows. 4. Особенности программирования библиотек в операционной системе Linux. 5. Особенности создания сервисных программ в операционных системах Windows и Linux. 6. Вызов библиотечных функций и их использование в программах. 7. Компиляция программы с включением в нее отладочной информации. 8. Межпроцессное взаимодействие программ. 9. Взаимодействие программных потоков. 	

2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

1. Назначение, состав и функции ОС.
2. Вычислительные процессы и ресурсы. Виды ресурсов.
3. Диаграмма состояний вычислительного процесса в ОС общего назначения и в ОС реального времени.
4. Понятие процесса Win32. Операции, выполняемые с процессами.
5. Мониторинг процессов и потоков Win32
6. Диспетчеризация потоков в Win32.
7. Потоки в Win32. Операции выполняемые с потоками.
8. Использование потоков при разработке приложений.
9. Квантование времени в MS Windows. Определение величины кванта времени.
10. Независимые и взаимодействующие потоки.
11. Понятие критического участка и критической секции.
12. Средства синхронизации потоков Win32.
13. Задача поставщик-потребитель и ее решение с помощью операций P(S) и V(S).
14. Объекты ядра MS Windows, их создание и использование.
15. Мьютексы, как средство синхронизации потоков.
16. Семафоры и их использование.
17. События и их использование.
18. Функции ожидания одного и нескольких событий
19. Сегментная и страничная адресация памяти. Сравнительный анализ
20. Deskрипторные таблицы. Схема вычисления физического адреса в режиме 286.
21. Формат дескрипторов в защищенных режимах 286 и 386.
22. Байт доступа. Поле доступа. Назначение битов.
23. Условие доступа программы к сегменту данных.
24. Реализация защиты ОС от прикладных программ.
25. Кольца защиты. Шлюзы вызова и задачи.
26. Аппаратная поддержка многозадачности. Сегмент состояния задачи.
27. Обработка прерываний в защищенном режиме.
28. Модель Холта.
29. Понятие тупика. Пример тупика процессов, использующих семафоры.
30. Необходимые условия возникновения тупика.
31. Методы борьбы с тупиками.
32. Управление страничной памятью. Стратегии подкачки и рабочие наборы страниц.
33. Разделы в виртуальном адресном пространстве процесса. Адресное пространство процесса MS Windows.
34. Управление памятью с помощью AWE.
35. Получение информации о состоянии виртуальной памяти
36. Резервирование регионов в адресном пространстве и передача физической памяти региону. Освобождение регионов
37. Кучи. Необходимость создания дополнительных куч. Операции с кучами.
38. Файлы, проецируемые в память
39. Динамически загружаемые библиотеки. Области применения.
40. Основные DLL MS Windows. Достоинства и недостатки DLL.
41. Использование DLL - импорт функций из DLL.
42. Явная и неявная загрузка DLL.
43. Эволюция архитектуры файловых систем.
44. Файловые системы Win32.
45. Структура магнитного диска.
46. Файловая система FAT.
47. Файловая система NTFS.
48. Основные отличия FAT и NTFS.

3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90 % от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76 % от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60 % от общего объёма заданных вопросов.

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

«Неудовлетворительно/не зачтено» – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*

- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

Критерии формирования оценок по зачету

«Отлично/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний, не допустил логических и фактических ошибок

«Хорошо/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний; допустил незначительные ошибки и неточности.

«Удовлетворительно/зачтено» – студент допустил существенные ошибки.

«Неудовлетворительно/не зачтено» – студент демонстрирует фрагментарные знания изучаемого курса; отсутствуют необходимые умения и навыки, допущены грубые ошибки.