МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВО «Кыргызско-Российский Славянский университет имени первого Президента Российской Федерации Б. Н. Ельцина»

Высшая школа цифровых технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Системы программирования»

Уровень высшего образования: МАГИСТРАТУРА

Направление подготовки: 09.04.04 (РФ) / 710400 (КР) -Программная инженерия

Магистерская программа: «Основы анализа и разработки приложений с большими объёмами распределённых данных»

Квалификация: магистр

Фонд оценочных средств предназначен для оценки результатов освоения дисциплины «Системы программирования» обучающимися магистратуры по направлению подготовки 09.04.04 «Программная инженерия»

Фонд оценочных средств рассмотрен и утвержден
на заседании Высшей школы Цифровых технологий протокол №1 от 02.09.26

Директор Высшей школы Цифровых технологий:

к.т.н., доцент Хмелева И.В.

Руководитель образовательной программы:

д.т.н., профессор Лыченко Н.М.

Исполнители (разработчики):

к.т.н., доцент Мансуров А.Н.

СОГЛАСОВАНО:

Зам. директора по учебной работе
Политехнического института КРСУ Комарцов Н.М.

Раздел 1. Перечень компетенций и этапы их формирования

Планируемые результаты обучения конкретизированы по разделу 3 РПД; формулировки компетенций и индикаторов взяты из РПД и ОПОП. Оценивание распределено по уровням A (знание), B (умение), C (владение) и D (промежуточная аттестация).

Компетенции и индикаторы	Планируемые результаты	Оценочные средства
ОПК-1. Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.	ЗНАТЬ: архитектуру вычислительных систем, трансляторы, загрузчики, форматы машинных и объектных программ. УМЕТЬ: применять алгоритмические методы при проектировании компонентов. ВЛАДЕТЬ: алгоритмизацией и экспериментальным исследованием системных средств.	A: теоретические вопросы; B: работы № 1–4; C: сборка и испытания; D: экзамен. ОПОП: ОПК-1.1–ОПК-1.3.
ОПК-3. Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.	ЗНАТЬ: организацию исходного и объектного кода, таблицы символов, промежуточные представления и интерфейсы компонентов. УМЕТЬ: анализировать техническую информацию и документировать архитектуру и результаты. ВЛАДЕТЬ: схемами, протоколами тестирования и сравнительным анализом.	A: разбор архитектуры; B: задания на интерфейсы; C: отчёты и аналитические заключения; D: экзамен. ОПОП: ОПК-3.1–ОПК-3.3.
ОПК-4. Способен применять на практике новые научные принципы и методы исследований.	ЗНАТЬ: современные методы реализации и исследования компиляторов, загрузчиков, эмуляторов и отладчиков. УМЕТЬ: применять и сопоставлять проектные решения и методы испытаний. ВЛАДЕТЬ: реализацией, экспериментальной проверкой прототипов и обоснованием решений.	A: вопросы по методам; B: алгоритмы компонентов; C: экспериментальные прототипы; D: экзамен. ОПОП: ОПК-4.1–ОПК-4.3.

Раздел 2. Технологическая карта дисциплины «Системы программирования»

Курс 1; семестр 1; трудоёмкость 5 ЗЕТ; промежуточная аттестация -экзамен.

Модуль/раздел	Контроль	Форма контроля	Мин.	Макс.	Срок
Модуль 1.Введение в основные понятия и компоненты систем программирования	Текущий	Анализ назначения и состава системы; сравнительная таблица инструментов	4	7	нед. 3
	Рубежный	Устный опрос по составу систем программирования и их назначению	3	5	нед. 3
Модуль 2. Раздел 2. Условная система программирования: состав, базовые соглашения и интерфейсы	Текущий	Защита практической работы № 4 «Эмулятор целевой ЭВМ»	4	7	нед. 6
	Рубежный	Задача на форматы данных, ISA, интерфейсы компонентов учебной системы	3	5	нед. 6
Модуль 3. Раздел 3. Компилятор с языка низкого уровня (Ассемблера)	Текущий	Защита практической работы № 2 «Компилятор с Ассемблера»	5	8	нед. 9
	Рубежный	Контрольная работа: двухпроходная трансляция, таблица символов, обработка ошибок	3	6	нед. 9
Модуль 4. Раздел 4. Загрузчики, статическая и динамическая компоновка	Текущий	Защита практической работы № 3 «Абсолютный загрузчик»	5	8	нед. 12
	Рубежный	Контрольная задача: загрузка объектного модуля, релокация, связывание	3	6	нед. 12
Модуль 5. Раздел 5. Компилятор с языка высокого уровня и проверка системы программирования	Текущий	Защита практических работ № 1 «Компилятор с ЯВУ» и № 5 «Отладчик»	6	11	нед. 16
	Рубежный	Сквозной тест учебной системы, протокол испытаний и ответы на вопросы	4	7	нед. 16
ВСЕГО за семестр		Текущий и рубежный контроль	40	70	нед. 3–16

Модуль/раздел	Контроль	Форма контроля	Мин.	Макс.	Срок
Промежуточная аттестация		Экзамен	20	30	по расписанию
СЕМЕСТРОВЫЙ РЕЙТИНГ		Сумма всех видов контроля	60	100	

Раздел 3. Типовые контрольные задания и оценочные материалы

Оценочные задания привязаны к разделам РПД и дифференцированы по уровню: A - воспроизведение и объяснение теории; B -применение способов решения; C -комплексное моделирование, реализация и документирование; D -промежуточный контроль.
Индивидуальные варианты допускают изменение исходных данных без изменения измеряемых результатов обучения.

Блок A. Вопросы для устного опроса и текущего контроля

Разделы 1 и 2. Компоненты систем программирования и интерфейсы учебной системы

1. Назовите основные и дополнительные компоненты системы программирования.
2. Чем технологическая платформа отличается от целевой? Что такое кросс-компилятор?
3. Опишите цепочку ЯВУ → ЯНУ → объектный модуль → загрузка → исполнение.
4. Как определить функционально полный состав учебной системы программирования?
5. Какие данные передаются по интерфейсам «компилятор -Ассемблер -загрузчик»?
6. Какие параметры ISA необходимо задать для учебной целевой ЭВМ?
7. Чем абсолютный объектный код отличается от перемещаемого?
8. Для чего служат эмулятор, отладчик, профилировщик?

Раздел 3. Компилятор с Ассемблера

1. Почему двухпроходный Ассемблер удобен для разрешения прямых ссылок?
2. Какие сведения содержатся в таблице символов?
3. Что делает первый проход Ассемблера? Что делает второй?
4. Как обрабатываются метки, директивы и операнды?
5. Как выявить повторное определение и неопределённый символ?
6. Как организовать исходную карту строк и сообщения об ошибках?
7. Как связаны синтаксис ЯНУ и формат объектной записи?

Раздел 4. Загрузчики, компоновка

1. Что делает абсолютный загрузчик? Какие проверки он обязан выполнять?
2. Чем компоновка отличается от загрузки?
3. Что такое релокация и таблица исправлений?
4. Как разрешаются внешние символы при связывании?

5. Сравните статическую и динамическую компоновку.

6. Какие преимущества и риски связаны с оверлейной загрузкой?

7. Как составить тестовые данные, выявляющие ошибки адресации?

Раздел 5. Компилятор ЯВУ и системные инструменты

1. Назовите этапы компиляции ЯВУ: лексический, синтаксический, семантический анализ, генерация кода.

2. Чем токен отличается от лексемы? Каковы лексические ошибки?

3. Как грамматика и дерево разбора соотносятся с абстрактным синтаксическим деревом (AST)?

4. Назовите методы синтаксического анализа сверху вниз и снизу вверх.

5. Как используются таблицы символов на этапе семантического анализа?

6. Что такое промежуточное представление (IR), для чего оно нужно?

7. Как реализовать генерацию инструкций для присваивания и арифметического выражения?

8. Какие этапы проходят при запуске программы на эмуляторе?

9. Как реализовать пошаговое исполнение и точку останова?

10. Чем различаются тесты лексера, парсера, генератора и сквозные тесты?

11. Как оформлять протокол испытания компонента?

12. Какие выводы можно сделать по анализу ошибок и производительности прототипа?

Единые соглашения для программных оценочных заданий

В учебном задании допускается собственная ISA. Для сопоставимости вариантов рекомендуется: регистры R0, R1 и PC; память M[0..255]; инструкции LOADI R,#n, LOAD R,[a], STORE [a],R, ADD R,#n, ADD R,R, SUB R,#n, JMP a, JZ R,a, OUT R, HALT; одна инструкция занимает одну ячейку. Формат текстового объектного модуля: «адрес | код операции | аргументы». Адресация и соглашения об операндах фиксируются в отчёте.

Типовые задания для контроля самостоятельной работы

Набор 1. Архитектура и интерфейсы

1. Построить схему инструментальной цепочки для ЯВУ→Ассемблер→объектный файл→абсолютный загрузчик→эмулятор; определить форматы минимум двух интерфейсов.

2. Для учебной машины с M[0..255] и командами LOADI, ADD, STORE, HALT показать, как размещается программа длиной 4 ячейки.

3. Сравнить состав системы при host=Windows/x86-64 и target=учебная 8-битная машина.

Набор 2. Ассемблер и таблицы

1. Для START: LOADI R0,#2; JMP END; WORD 99; END: ADD R0,#5; HALT построить таблицу символов и адреса инструкций при начальном адресе 0.

2. Предложить структуру промежуточной записи первого прохода и алгоритм разрешения прямой ссылки END.

3. Предъявить по два положительных и отрицательных теста компилятора с ЯНУ.

Набор 3. Загрузка и исполнение

1. Описать загрузку записей 0|LOADI|R0,#3; 1|ADD|R0,#4; 2|STORE|[20],R0; 3|HALT|-в память.

2. Выполнить трассировку PC и R0, отметить состояние M[20].

3. Объяснить отличие абсолютной загрузки от перемещаемой для тех же записей.

Набор 4. ЯВУ и отладка

1. Для x=2+3; print(x); выделить токены, построить AST и составить набор команд целевой машины.

2. Предложить диагностику ошибок x=2+; и print(y); при отсутствии присваивания y.

3. Составить тест-кейс остановки на STORE и наблюдения значения ячейки 20.

Блоки B и C. Практические задания

Каждая работа завершается защитой: постановка задачи, исходные данные и формальная модель / алгоритм, результаты, проверки, выводы. Преподаватель может предложить эквивалентный вариант с иными обозначениями и параметрами.

Практическая работа № 4. Эмулятор целевой ЭВМ

Раздел 2; 4 часа

1. Сформулировать описание ISA учебной машины: набор команд, регистры R0, R1, PC, память M[0..255], правила адресации; зафиксировать способы задания констант и переходов.

2. Реализовать цикл fetch–decode–execute; исполнить программу LOADI R0,#3; ADD R0,#4; STORE [20],R0; HALT.

3. Добавить диагностические события «неизвестная команда», «адрес вне диапазона», «превышен лимит шагов».

4. Проверить состояние регистров и памяти после каждой инструкции; измерить число выполненных шагов.

Представляемый результат: Функционирующий эмулятор, листинг ISA, журнал выполнения; эталон: после STORE M[20]=7.

Практическая работа № 2. Компилятор с Ассемблера (ЯНУ)

Раздел 3; 6 часов

1. Задать текстовый формат Ассемблера: LABEL: OPCODE operand; комментарии и директивы ORG, WORD; определить допустимые имена.

2. Реализовать первый проход (счётчик адресов, таблица символов) и второй проход (разрешение меток, выдача объектных записей).

3. Тест: START: LOADI R0,#2; JMP END; WORD 99; END: ADD R0,#5; HALT. Построить таблицу символов и объектный модуль.

4. Подготовить отрицательные тесты: дублирующаяся метка, неопределённая метка, неподдерживаемая команда, неверный операнд.

Представляемый результат: Файлы исходного текста, таблица символов, объектный файл, журнал обоих проходов, протокол ошибок.

Практическая работа № 3. Абсолютный загрузчик

Раздел 4; 4 часа

1. Определить формат объектной записи «адрес|код|операнды» и проверку адресного диапазона 0..255.

2. Загрузить программу по адресам 0..3 и отобразить карту заполненных ячеек; проверить начальное значение PC.

3. Провести отрицательные тесты: выход за границы, дублирование адреса, некорректный opcode, повторная загрузка без очистки памяти.

4. Сравнить свойства абсолютной загрузки и релокации на небольшом примере.

Представляемый результат: Модуль загрузчика, карта памяти, результаты тестов, описание ограничений абсолютной схемы.

Практическая работа № 1. Компилятор с ЯВУ

Раздел 5; 6 часов

1. Определить входной мини-язык с операторами x = expression; и print(expression);, целыми литералами, именами, +, − и скобками.

2. Реализовать лексический анализатор с указанием типов токенов и координат ошибки; описать грамматику и построить AST.

3. Реализовать семантическую проверку объявлений/присваиваний и генерацию команд учебной ISA. Тестовая программа: x=2+3; print(x);.

4. Сравнить по тестам: валидную программу, неверный символ, пропуск «;», неизвестную переменную; показать форму диагностик.

Представляемый результат: Таблица токенов, дерево разбора/AST, объектный код и результаты исполнения; эталон для основной программы: печать 5.

Практическая работа № 5. Отладчик кода эмулятора

Раздел 5; 6 часов

1. Реализовать команды пошагового выполнения, продолжения, установки/снятия точек останова, просмотра PC, регистров и области памяти.

2. На тестовой программе LOADI R0,#3; ADD R0,#4; STORE [20],R0; HALT установить breakpoint по адресу 2.

3. Подтвердить, что до исполнения STORE состояние M[20] неизменно, а после одного шага равно 7.

4. Проверить реакцию на недопустимый breakpoint, завершение программы и невозможность продолжения после HALT.

Представляемый результат: Рабочий отладчик, скриншоты/журнал двух режимов, таблица ожидаемых и наблюдаемых результатов.

Контрольные ориентиры к заданиям (для преподавателя)

1. Работа № 4: трасса LOADI R0,#3 → R0=3; ADD R0,#4 → R0=7; STORE [20],R0 → M[20]=7; HALT завершает программу. При одноячеечных командах PC увеличивается на 1, если не выполнен переход.

2. Работа № 2: при ORG=0 метки START=0, END=3; JMP END должен ссылаться на адрес 3. Директива WORD занимает одну ячейку. Учебный формат кода фиксируется в спецификации.

3. Работа № 1: для программы x=2+3; print(x); при корректной таблице переменных и генерации команд выводится число 5; пропущенная точка с запятой -синтаксическая ошибка, а чтение неизвестной у -семантическая ошибка.

4. Работа № 5: при останове перед STORE по PC=2 регистр R0 равен 7, а M[20] ещё не изменена; после выполнения STORE M[20]=7.

5. Отличие загрузки/компоновки: абсолютный загрузчик размещает модуль по фиксированным адресам; релокация предполагает изменение адресных ссылок при перемещении; компоновка разрешает связи между модулями.

Блок D. Промежуточная аттестация

Форма: экзамен. Билет включает теоретические вопросы и практические задания; результаты оценивания показывают отдельно достижение уровней «знать» и «уметь/владеть». Представленные ниже билеты типовые и могут быть дополнены вариантами с эквивалентной сложностью.

Билет № 1

Вопросы («ЗНАТЬ»):

1. Состав систем программирования. Host и target.

2. Двухпроходная архитектура Ассемблера.

3. Статическая и динамическая компоновка.

Задания («УМЕТЬ» и «ВЛАДЕТЬ»):

1. Для текста START: LOADI R0,#2; JMP END; WORD 9; END: HALT построить таблицу символов и адреса.

2. Определить последовательность состояний R0, PC для LOADI R0,#3; ADD R0,#4; HALT.

Билет № 2

Вопросы («ЗНАТЬ»):

1. Этапы компиляции ЯВУ.

2. Форматы объектных модулей и абсолютная загрузка.

3. Назначение и интерфейсы отладчика.

Задания («УМЕТЬ» и «ВЛАДЕТЬ»):

1. Выделить токены и построить AST для x=2+3; print(x);.

2. Загрузить объектные записи по адресам 0,1,2 и описать обработку записи с адресом 256.

Билет № 3

Вопросы («ЗНАТЬ»):

1. Организация таблицы символов и её роль.

2. Алгоритмы лексического и синтаксического анализа.

3. Протокол экспериментальной проверки системных компонентов.

Задания («УМЕТЬ» и «ВЛАДЕТЬ»):

1. Предложить диагностику ошибок при неверном символе @ в программе x=2@3; и неизвестном идентификаторе y.

2. Разработать четыре тест-кейса для отладчика (breakpoint, step, inspect, завершение) с ожидаемым результатом.

Раздел 4. Методические материалы и критерии оценивания

Контроль проводится по представленным в разделе 3 оценочным материалам. Баллы начисляются за проверяемые результаты, а не за факт присутствия. При оценке письменной работы учитываются полнота решения, корректность выбранного формального/алгоритмического подхода, обоснованность выводов и воспроизводимость результатов.

4.1. Показатели текущего контроля и защиты практических работ

Компонент оценки практической работы	Доля в оценке	Критерий
Корректность модели/алгоритма/программного решения	40 %	Соответствие постановке и правильность содержательных результатов.
Проверка на тестах или контрпримерах	25 %	Наличие тестов, проверка исключений, корректная интерпретация результата.
Отчёт и воспроизводимость	20 %	Достаточность данных, схем, листингов и протоколов.
Защита и ответы на вопросы	15 %	Самостоятельное объяснение метода и причин полученных результатов.

Для каждой контрольной точки преподаватель переводит процент выполнения в баллы: начисленные баллы = максимальный балл точки × доля выполненных критериев / 100 с округлением по принятой в вузе процедуре. Невыполненная/непредставленная работа оценивается 0 баллами.

4.2. Уровни освоения на текущем и рубежном контроле

Диапазон выполнения	Характеристика
85–100 %	Полное понимание, верные результаты, обоснованные выводы и самостоятельная защита.
70–84 %	Существенно верное решение при незначительных неточностях.

60–69 %	Основные действия выполнены, но есть частные ошибки/пробелы.
31–59 %	Существенные ошибки; многие требования к заданию не выполнены.
0–30 %	Не достигнут минимальный результат или отсутствует содержательное решение.

4.3. Рубежный контроль

Рубежные контрольные точки в технологической карте представлены проверкой самостоятельной работы: архитектурных схем, таблиц символов, интерфейсных описаний, наборов тестов и аналитических выводов. Это не вводит отдельной курсовой работы или дополнительной формы промежуточной аттестации.

4.4. Промежуточная аттестация

Промежуточная аттестация -экзамен; максимум 30 баллов. Рекомендуемая структура билета: теоретическая часть -10 баллов; два практических задания -по 10 баллов. Вопросы и задания преподаватель выбирает с учётом охвата основных разделов РПД.

Элемент билета	Баллы	Проверяемые показатели
Теоретические вопросы (совокупно)	0–10	Точность определений и понятий (4), полнота связей и аргументации (4), корректный пример (2).
Практическое задание 1	0–10	Корректность постановки и метода (3), правильность решения (5), объяснение результата (2).
Практическое задание 2	0–10	Корректность постановки и метода (3), правильность решения (5), объяснение результата (2).

Для аттестации преподаватель фиксирует результаты всех частей билета в оценочном листе. Предлагаемое время подготовки и выполнения -до 80 минут; фактический регламент устанавливается расписанием и локальными документами вуза.

4.5. Перевод суммы баллов в оценку

Семестровый рейтинг	Оценка
85–100	Отлично
70–84	Хорошо
60–69	Удовлетворительно
0–59	Неудовлетворительно / не зачтено

Рейтинг складывается из баллов по всем мероприятиям технологической карты (не более 70) и результатам промежуточной аттестации (не более 30). Шкала воспроизводит логику предоставленного образца ФОС; при необходимости должна быть сверена с действующим локальным положением КРСУ.

Раздел 5. Методические рекомендации для обучающихся

5.1. Подготовка к занятиям и текущему контролю

- Перед реализацией прототипа зафиксировать спецификацию ISA, форматы входа/выхода, интерфейсы и ограничения окружения.
- Сохранять трассу прохождения данных: исходный код → токены/AST → Ассемблер → объектные записи → загрузка → исполнение.
- Для двухпроходного Ассемблера сохранять таблицу символов, состояния счётчика адресов и сообщения об ошибках каждого прохода.
- Раздельно тестировать лексер, парсер, генератор кода, загрузчик, эмулятор и отладчик, а затем выполнять сквозной тест.
- Сравнивать ожидаемые и наблюдаемые результаты в таблице; фиксировать ограничения прототипа и воспроизводимые дефекты.

5.2. Требования к отчёту по практической работе

1. Титульные сведения: дисциплина, номер работы, тема, цель, Ф. И. О. исполнителя, группа, дата.

2. Постановка задачи, исходные данные и индивидуальный вариант.

3. Описание используемой модели/алгоритма; схемы, таблицы, формулы или листинги.

4. Пошаговое решение либо описание программного прототипа и условий выполнения.

5. Тестирование и протокол: входные данные, ожидаемый результат, полученный результат, заключение.

6. Вывод о достижении цели работы, перечень выявленных ограничений.

5.3. Подготовка к промежуточной аттестации

Повторить вопросы блока А, самостоятельно прорешать задачи блока В и комплексные задания блока С. Для каждого раздела уметь не только привести определения, но и выполнить формальную проверку, построить контрпример или продемонстрировать результаты исполнения. Во время ответа явно указывать допущения, аргументировать каждый шаг и проверять правильность результата.

5.4. Самостоятельная работа

Самостоятельная работа соответствует тематике и объёму, указанным в РПД. Допускаются учебные проекты, мини-эксперименты и дополнительные варианты, если они не меняют обязательные формы текущего и промежуточного контроля. Все материалы оформляются с указанием используемых источников.