

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Физико-технический институт

Факультет информатики и вычислительной техники

Кафедра программного обеспечения вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой ПОВТ



С.Г. Федорченко

«28» августа 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине (модулю)

Б1.В.ДВ.04.01 МАШИННО-ЗАВИСИМЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

на 2023/2024 учебный год

Направление

2.09.03.04 Программная инженерия

Профиль

Разработка программно-информационных систем

Квалификация

бакалавр

Форма обучения

очная, заочная

ГОД НАБОРА 2022

доцент кафедры ПОВТ,



С.Г. Федорченко

«28» августа 2023 г.

Тирасполь, 2023

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Машинно-зависимые языки программирования» у обучающегося должны быть сформированы следующие компетенции:

Задача ПД	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции
Обязательные профессиональные компетенции выпускников и индикаторы их достижения		
Тип задач профессиональной деятельности: <i>проектный</i>		
Формирование требований к информатизации и автоматизации прикладных процессов, формализация предметной области проекта; технико-экономическое обоснование проектных решений и составление технического задания на разработку программного продукта; проектирование программно-аппаратных средств в соответствии с техническим заданием; применение современных инструментальных средств при разработке программного обеспечения; документирование компонентов информационной системы на стадии жизненного цикла	ПК-7. Способность оценивать временную и емкостную сложность программного обеспечения	ИД-1 _{ПК-7} Знает методы оценки временной и емкостной сложности программного обеспечения ИД-2 _{ПК-7} Умеет вычислять временную и емкостную сложность ПО ИД-3 _{ПК-7} Имеет навыки оценки временной и емкостной сложности ПО
Тип задач профессиональной деятельности: <i>производственно-технологический</i>		
Проведение работ по установке программного обеспечения автоматизированных систем и загрузки баз данных; настройка параметров ИС и тестирование результатов настройки; ведение технической документации; техническое сопровождение	ПК-10. Владение навыками использования различных технологий разработки программного обеспечения	ИД-1 _{ПК-10} Знает современные технологии разработки ПО (структурное, объектно-ориентированное) ИД-2 _{ПК-10} Умеет использовать современные технологии разработки ПО ИД-3 _{ПК-10} Имеет навыки использования современных технологий разработки ПО

Задача ПД	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции
Обязательные профессиональные компетенции выпускников и индикаторы их достижения		
Тип задач профессиональной деятельности: <i>проектный</i>		
ИС в процессе эксплуатации; применение <i>Web</i> технологий при реализации удаленного доступа в системах клиент-сервер и распределенных вычислений		

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины, их название	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
РУБЕЖНЫЙ КОНТРОЛЬ	Раздел 1. Архитектура ЭВМ и микропроцессоров <i>Intel</i>	ПК-7, ПК-10	Контрольная работа №1
РУБЕЖНАЯ АТТЕСТАЦИЯ	Раздел 2. Операторы языка Ассемблер		Лабораторная работа №1 Лабораторная работы №2 Лабораторная работы №3 Лабораторная работы №4 Лабораторная работы №5 Контрольная работа №1 Контрольная работа №2
	Раздел 3. Работа с файлами в языке Ассемблер		Лабораторная работа №6 Контрольная работа №2
	Раздел 4. Работа с сопроцессором		Контрольная работа №2
	Раздел 4. Работа в защищенном режиме		Лабораторная работы №7 Лабораторная работы №8
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
№1		ПК-7, ПК-10	Экзамен

3. Показатели и критерии оценивания компетенции по этапам формирования, описание шкал оценивания

Этапы оценивания компетенции	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
Первый этап	ИД-1ПК-7 Знает методы оценки временной и емкостной сложности программного обеспечения	Не знает	Знает методы оценки временной и емкостной сложности программного обеспечения, но затрудняется с их	Знает методы оценки временной и емкостной сложности программного обеспечения, но делает ошибки,	Знает методы оценки временной и емкостной сложности программного обеспечения. Умеет адекватно их применять

Этапы оценивания компетен-	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
			применением в реальных задачах	не влияющие на результаты	
Второй этап	ИД-2ПК-7 Умеет вычислять временную и емкостную сложность ПО	Не умеет	Умеет вычислять временную и емкостную сложность ПО, но допускает много ошибок при их применении для решения профессиональных задач	Умеет вычислять временную и емкостную сложность ПО, но выбирает не оптимальный путь решения задачи	Умеет вычислять временную и емкостную сложность ПО
Третий этап	ИД-3ПК-7 Имеет навыки оценки временной и емкостной сложности ПО	Не владеет	Не в полной мере владеет навыками оценки временной и емкостной сложности ПО	Владеет навыками оценки временной и емкостной сложности ПО, но допускает не критические ошибки	Владеет навыками оценки временной и емкостной сложности ПО
Первый этап	ИД-1ПК-10 Знать современные технологии разработки ПО (структурное, объектно-ориентированное)	Не знает	Знает современные технологии разработки ПО, но затрудняется с их применением в реальных задачах	Знает современные технологии разработки ПО, но делает ошибки, не влияющие на результаты	Знает современные технологии разработки ПО. Умеет адекватно их применять
Второй этап	ИД-2ПК-10 Уметь использовать современные технологии разработки ПО	Не умеет	Умеет использовать современные технологии разработки ПО, но допускает много ошибок при их применении для решения профессиональных задач	Умеет использовать современные технологии разработки ПО, но выбирает не оптимальный путь решения задачи	Умеет использовать современные технологии разработки ПО
Третий этап	ИД-3ПК-10 Иметь навыки использования современных технологий разработки ПО	Не владеет	Не в полной мере владеет использованием современных технологий разработки ПО	Владеет навыками использования современных технологий разработки ПО, но допускает не критические ошибки	Владеет использованием современных технологий разработки ПО

4. Шкала оценивания

Согласно Положению «О порядке организации аттестации в ИТИ ПГУ им. Т.Г. Шевченко, итоговая оценка представляет собой сумму баллов, полученных студентом по итогу освоения дисциплины (модуля):

Оценка в традиционной шкале	Оценка в 100-балльной шкале	Буквенные эквиваленты оценок в шкале ЗЕ (% успешно аттестованных)
5 (отлично)	88–100	А (отлично) – 88-100 баллов
4 (хорошо)	70–87	В (очень хорошо) – 80-87баллов
		С (хорошо) – 70-79 баллов
3 (удовлетворительно)	50–69	Д (удовлетворительно) – 60-69 баллов
		Е (посредственно) – 50-59 баллов
2 (неудовлетворительно)	0–49	Гх – неудовлетворительно, с возмож- ной пересдачей – 21-49 баллов
		Г – неудовлетворительно, с повтор- ным изучением дисциплины – 0-20 баллов

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице, указанной ниже

А	“Отлично” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.
В	“Очень хорошо” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
С	“Хорошо” - теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
Д	“Удовлетворительно” - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
Е	“Посредственно” - теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному.
ГХ	“Условно неудовлетворительно” - теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий.
Г	“Безусловно неудовлетворительно” - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий.

5. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы

формирования компетенций при изучении учебной дисциплины в процессе освоения образовательной программы

5.1 Типовой вариант лабораторной работы № 1.

ЗАДАНИЕ к лабораторной работе № 1

Разработать программу, выдающую на экран приветствие в адрес студентов, поступивших в ИТИ. Привести список дисциплин, изучаемых в первом семестре.

5.2 Типовой вариант лабораторной работы № 2.

ЗАДАНИЕ к лабораторной работе № 2

Выполнить вычисления, в соответствии с полученным заданием. Выдать на экран полученные результаты.

5.3 Типовой вариант лабораторной работы № 3.

ЗАДАНИЕ к лабораторной работе № 3

Заработать программу, которая:

- 1) вычисляет значение заданного арифметического выражения;
- 2) вычисляет значение заданного логического выражения;
- 3) вычисляет значение заданного арифметического выражения с помощью, преимущественно, операций сдвига.

Исходные значения переменных задать в программе. Вычисление каждого пункта программы оформить в виде процедуры: а) без параметров; б) с параметрами, через которые передаются значения переменных

5.4 Типовой вариант контрольной работы № 1.

Ответьте на следующие вопросы:

1. Как организовано взаимодействие процессора и оперативной памяти, что такое кеш-память?
2. Адресация памяти в реальном режиме работы микропроцессора.
3. Что такое прерывание, вектор прерывания?

5.5 Типовой вариант лабораторной работы № 4.

ЗАДАНИЕ к лабораторной работе № 4

Заработать программу, которая:

- 4) вычисляет значение заданного арифметического выражения;
- 5) вычисляет значение заданного логического выражения;
- 6) вычисляет значение заданного арифметического выражения с помощью, преимущественно, операций сдвига.

Исходные значения переменных задать в программе. Вычисление каждого пункта программы оформить в виде макроса: а) без параметров; б) с параметрами, через которые передаются значения переменных

5.6 Типовой вариант лабораторной работы № 5.

Задание к лабораторной работе № 5

Написать программу, которая:

Разработать программу, которая реализует меню из 3-й пунктов. Название меню: 'Способы присоединения меню к окну'.

Названия пунктов – '1-й способ', '2-й способ', '3-й способ'.

Пользователь, выбирая один из них получает описание соответствующего способа.

5.7 Типовой вариант лабораторной работы № 6.

Задание к лабораторной работе № 6

Написать программу, которая:

- 1) вводит с клавиатуры текст;
- 2) записывает его в файл, имя файла: "text.txt";
- 3) если введена пустая строка – закрывает файл и прекращает работу программы.

Написать программу, которая:

- 1) читает содержимое файла "text.txt" и выводит его на экран;
- 2) когда достигнут конец файла закрывает его и выдает сообщение: "file text.txt is closed".

5.8 Типовой вариант лабораторной работы № 7

Задание к лабораторной работе № 7

Написать программу, которая последовательно:

- выводит на экран 1-е окно;
- после нажатия оператором кнопки ОК закрывает его;
- выводит на экран 2-е окно;
- после нажатия оператором кнопки ОК закрывает его;
- выводит на экран 3-е окно;
- после нажатия оператором кнопки ОК закрывает программу.

Параметры окон зависят от номера вашего варианта и приведены в таблице.

5.9 Типовой вариант лабораторной работы № 8

Задание к лабораторной работе № 8

Разработать программу, которая предлагает оператору угадать букву, позволяет ему много раз повторить попытку, и в конце говорит, с какого раза он угадал.

5.10 Типовой вариант контрольной работы № 2.

Ответить на следующие вопросы:

1. Что такое дескриптор ?
2. Как организовано обращение к памяти при ее страничной организации.
3. Механизм прерываний в Windows.

5.11 Типовой тест к экзамену

Тест

1. Принципы фон Неймана:
 - 1) линейное пространство памяти;
 - 2) принцип предсказуемости;
 - 3) принцип микропрограммирования;
 - 4) принцип обязательности.
2. Сегмент, это:
 - 1) область памяти, размером не более 64 К;
 - 2) непрерывная область памяти, произвольного размера;
 - 3) непрерывная область памяти, размером не более 64 К;
 - 4) область памяти произвольного размера.
3. Конвейер, позволяет:
 - 1) увеличить точность вычислений;
 - 2) организовать параллельное выполнение нескольких команд;
 - 3) реализовать многоядерный процессор;
 - 4) разбить исполнение команды на несколько стадий;

- 5) уменьшить время, необходимо на выполнение действия над дробными числами.
4. SEGMENT, это:
- 1) команда описания сегмента;
 - 2) директива описания сегмента;
 - 3) директива, показывающая закрепление сегментных регистров;
 - 4) директива, задающая порядок объединения сегментов;
 - 5) команда, задающая порядок объединения сегментов.
5. Com – программа:
- 1) занимает несколько сегментов;
 - 2) занимает максимум 64К памяти;
 - 3) в отдельном сегменте находятся данные, в отдельном-команды;
 - 4) в одном сегменте расположены и команды и данные;
 - 5) требует описание стека.
6. Выражение b1 db 10 описывает:
- 1) переменную b1 занимающую 1 слово;
 - 2) переменную b1 занимающую 1 байт;
 - 3) переменную b1 занимающую 1 двойное слово;
 - 4) переменную b1 занимающую 1 6 байт;
 - 5) переменную b1 занимающую 10 байт.
7. Команда mov:
- 1) требует одинаковых размеров источника и приемника;
 - 2) требует одинаковых типов данных источника и приемника;
 - 3) пересылает данные из источника в приемник;
 - 4) пересылает данные из приемника в источник.
8. Команда add позволяет:
- 1) сложить две величины;
 - 2) вычесть две величины;
 - 3) требует одинаковых размеров операндов;
 - 4) не изменить флаги.
9. Команда shl ax,2:
- 1) заносит «лишний» бит в CF;
 - 2) сдвигает вправо содержимое ax на 2 разряда;
 - 3) сдвигает влево содержимое ax на 2 разряда;
 - 4) не меняет CF;
 - 5) меняет IF.
10. Выполняется следующий набор команд
- ```
mov ax, 01000100b
and ax, 00001111b
```
- в результате регистр ax содержит:
- 1) 01000100b;
  - 2) 00000100b;
  - 3) 01000000b;
  - 4) 01000101b.
11. Команда je m1:
- 1) передает управление на метку m1;
  - 2) не передает управление на метку m1;
  - 3) передает управление на метку m1, если команда сравнения ранее подтвердила равенство сравниваемых величин;



- 4) не передает управление на метку m1, если команда сравнения ранее подтвердила равенство сравниваемых величин;
- 5) передает управление на метку m1, если команда сравнения ранее подтвердила что сравниваемая величина>0;

12. Дан следующий набор команд:

```
d1 db 256 dup(?)
d2 db 256 dup(?)
 mov si,offset d1
 mov di,offset d2
 mov cx,256
 cld
 rep cmpsb
```

В результате их выполнения:

- 1) произойдет сравнение строк;
- 2) произойдет пересылка строк;
- 3) произойдет заполнение d1 значением 0;
- 4) операция сравнения повторится 256 раз.

13. Дан следующий набор команд:

```
mov ax,2
mov bx,2
cmp ax,bx
je m1
jg m2
jl m3
```

В результате их выполнения:

- 1) будет передано управление на метку m1
- 2) будет передано управление на метку m2
- 3) будет передано управление на метку m3
- 4) не будет передано управление на указанные метки.

14. Дан следующий набор команд:

```
mov cx,6
 mov bx,0
m1: inc bx
 loop m1
```

После их выполнения, чему будет равен регистр bx:

- 1) 1;
- 2) 6;
- 3) 5;
- 4) 0.

15. Команда PUSHF:

- 1) обнуляет стек;
- 2) заполняет стек содержимым регистров AX,CX,DX,BX,SP,BP,SI,DI;
- 3) считывает из стека содержимое в регистры AX,CX,DX,BX,SP,BP,SI,DI;
- 4) заполняет стек содержимым регистров AX,CX,DX,BX;
- 5) считывает из стека содержимое в регистры AX,CX,DX,BX.

16. Дан набор команд:

```
P1 proc
 push ax

```

```
 pop ax
P1 endp
P2 proc
.....
 ret
P2 endp
```

Тестирование показало, что работа процедуры P1 неверна, найдите причину этого:

- 1) неверно записана процедура P2;
- 2) неверно организована работа со стеком в процедуре P1;
- 3) отсутствует оператор `ret` в процедуре P1;
- 4) присутствует оператор `ret` в процедуре P2.

17. Макроопределения в языке ассемблер:

- 1) позволяют уменьшить конечный код программы;
- 2) позволяют увеличить читабельность исходного кода программы;
- 3) позволяют увеличить скорость работы программы;
- 4) увеличивают конечный код программы.

18. Команда `int` позволяет:

- 1) организовать программное прерывание;
- 2) организовать аппаратное прерывание;
- 3) вызвать обработчик прерывания;
- 4) запрещает прерывание.

19. Для организации хранения информации на магнитном диске используются:

- 1) дорожки и сектора;
- 2) кластеры;
- 3) стек;
- 4) сегменты;
- 5) флаги.

20. Для работы с файлом, используя прерывание `21h` необходимо:

- 1) открыть файл;
- 2) выполнить нужные действия над файлом;
- 3) закрыть файл;
- 4) прочитать сектор;
- 5) записать сектор.

21. Сопроцессор позволяет:

- 1) ускорить работу основного процессора;
- 2) выполнить арифметические операции над дробными числами;
- 3) организовать работу с файлами;
- 4) организовать работу с видеопамятью.

22. Стек сопроцессора:

- 1) имеет фиксированный размер;
- 2) имеет переменный размер;
- 3) может хранить данные переменной длины;
- 4) хранит данные только одной длины.

23. Форматы данных, с которыми работает сопроцессор:

- 1) строки;
- 2) нечисла;
- 3) вещественные числа;
- 4) изображения.

24. Команда `fld s1`:

- 1) загружает вещественное число `s1` в вершину стека сопроцессора;

- 2) загружает целое число s1 в вершину стека сопроцессора;
- 3) записывает вещественное число в s1 из вершины стека сопроцессора;
- 4) записывает целое число в s1 из вершины стека сопроцессора.

25. Команда fsqrt позволяет:

- 1) вычислить абсолютное значение от числа, находящегося в вершине стека сопроцессора;
- 2) округлить до целого значения число, находящееся в вершине стека сопроцессора;
- 3) вычислить квадрат значения, находящегося в вершине стека сопроцессора;
- 4) вычислить квадратный корень из значения, находящегося в вершине стека сопроцессора.

Ответы на тест по предмету «Программирование на языке низкого уровня»:

1. -1,3
2. -3
3. - 2,4.
4. - 2,4
5. - 2,4
6. - 2
7. - 1,3
8. - 1,3
9. - 1,3
10. - 2.
11. - 3.
12. - 1.
13. - 1.
14. - 2
15. - 2
16. - 3
17. - 2,3,4
18. - 1,3
19. - 1,2
20. - 1,2,3
21. - 2
22. -1,4
23. -2,3.
24. - 1.
25. - 3.

#### 5.12 Примерная тематика курсовых работ

Не планируются

### **Вопросы к экзамену по дисциплине «Машинно-зависимые языки программирования»**

1. Особенности программирования на Ассемблере.
2. Основные сведения об архитектуре процессора Intel.
3. Понятие конвейеризации, роль кэш.
4. Организация памяти в реальном режиме
5. Режимы работы микропроцессора.
6. Регистры микропроцессора.
7. Типы данных. Форматы команд
8. Этапы создания программ на языке ассемблер.
9. Структура программы, особенности программ в COM файлах
10. Структура программы, особенности программ в EXE файлах

11. Режимы адресации данных в реальном режиме
12. Команды сравнения
13. Команды перехода
14. Работа с процедурами
15. Понятие о прерывании. Вектор прерывания. Обработка прерывания
16. Создание консольных приложений в защищенном режиме
17. Работа со строками в защищенном режиме
18. Работа с процедурами в защищенном режиме
19. Организация циклов в программе. Вложенные циклы.
20. Команды сдвига.
21. Команды логических операций.
22. Арифметические команды.
23. Арифметические команды в ASCII формате.
24. Команды обработки строк.
25. Сложные структуры данных. Массивы.
26. Сложные структуры данных. Структуры.
27. Сложные структуры данных. Объединения, записи
28. Макросредства языка Ассемблер
29. Организация дисковой памяти. Оглавление диска. FAT.
30. Физическая структура NTFS
31. Передача параметров в процедуру
32. Соглашения cdecl и stdcall
33. Архитектура сопроцессора.
34. Сопроцессор, его функции
35. Стек сопроцессора
36. Форматы данных сопроцессора
37. Взаимодействие основного процессора и сопроцессора
38. Исключения сопроцессора и их обработка
39. Специальные численные значения сопроцессора
40. Мнемоническое обозначение команд сопроцессора
41. Понятие дескриптора
42. Системные регистры МП, используемые в защищенном режиме
43. Регистры отладки МП
44. Особенности защищенного режима микропроцессора
45. Таблица GDT
46. Таблица LDT
47. Адресация памяти в защищенном режиме
48. Организация исполнения нескольких программ в защищенном режиме
49. Страничная трансляция адресов
50. Организация прерываний в защищенном режиме