

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Системное программирование»
для студентов направления подготовки
09.03.01 Информатика и вычислительная техника
Направленность (профиль)

«Программное обеспечение средств вычислительной техники и автоматизированных систем»

Ставрополь 2026

Лабораторная работа №1

Тема: Основы программирования на языке Assembler.

Учебные вопросы.

1. Программа типа COM.
2. Программа типа EXE.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 128-131.

Программа, написанная на ассемблере, так же как и программа, написанная на любом другом языке программирования, выполняется не сама по себе, а при помощи операционной системы. Операционная система выделяет области памяти для программы, загружает ее, передает ее управление и обеспечивает взаимодействие программы с устройствами ввода-вывода, файловыми системами и другими программами (кроме тех случаев, когда программа сама является операционной системой или ее частью).

Самая простая и распространенная операционная система для компьютеров, основанных на процессорах Intel, - DOS. DOS предоставляет программам полную свободу действий, никак не ограничивая доступ к памяти и внешним устройствам, позволяя им самим управлять процессором и распределением памяти.

Чтобы программа выполнялась любой ОС, она должна быть скомпилирована в исполняемый файл. Основные два формата исполняемых файлов в DOS - **COM** и **EXE**. Файлы типа COM содержат только скомпилированный код без какой-либо дополнительной информации о программе. Весь код, данные и стек такой программы располагаются в одном сегменте и не могут превышать 64 Кб. Файлы типа EXE содержат заголовок, где описывается размер файла, требуемый объем памяти, список команд в программе, использующих абсолютные адреса, которые зависят от расположения программы в памяти, и т.д. EXE-файл может иметь любой размер.

1. Программа типа COM

Традиционно первая программа для освоения нового языка программирования - программа, выводящая на экран текст Hello world!.

Наберите в любом текстовом редакторе, который может записывать файлы как обычный текст (например: EDIT.COM в DOS, встроенный редактор в Norton Commander или аналогичной программе, NOTEPAD в Windows), следующий текст:

```
;Hello-1.asm
;Выводит на экран сообщение "Hello World!" и завершается.
.model tiny                ;Модель памяти, используемая для COM.
.code                     ;Начало сегмента кода.
org 100h                  ;Начальное значение счетчика - 100h.
start:      mov ah,9       ;Номер функции DOS - в AH.
mov dx,offset message     ;Адрес строки - в DX.
int 21h                  ;Вызов системной функции DOS.
ret                      ;Завершение COM-программы.
message db "Hello World!",0Dh,0Ah,'$' ;Строка для вывода.
end start                ;Конец программы.
```

и сохраните его как файл hello-1.asm. Можно также использовать готовый файл с этим именем.

Для превращения программы в исполняемый файл сначала надо вызвать ассемблер, чтобы скомпилировать ее в объектный файл с именем hello-1.obj, набрав в командной строке следующую команду:

```
tasm hello-1.asm ,,,
```

А затем запустить компоновщик:

```
tlink /t /x hello-1.obj
```

В результате будет получен файл HELLO-1.COM размером 23 байта. Если его выполнить, на экране появится строка Hello World! и программа завершится.

Рассмотрим исходный текст программы, чтобы понять, как она работает.

Первая строка определяет модель памяти TINY, в которой сегменты кода, данных и стека объединены. Эта модель предназначена для создания файлов типа COM.

Директива .CODE начинает сегмент кода, который в нашем случае также должен содержать и данные.

ORG 100h устанавливает значение программного счетчика в 100h, потому что при загрузке COM-файла в память DOS занимает первые 256 байт (100h) блоком данных PSP и располагает код программы только после этого блока. Все программы, которые компилируются в файлы типа COM, должны начинаться с этой директивы.

Метка START располагается перед первой командой в программе и будет использоваться в директиве END, чтобы указать, с какой команды начинается программа.

Команда MOV AH,9 помещает число 9 в регистр AH - номер функции DOS «вывод строки».

Команда MOV DX,OFFSET MESSAGE помещает в регистр DX смещение метки MESSAGE относительно начала сегмента данных, который в нашем случае совпадает с сегментом кода.

Команда INT 21h вызывает системную функцию DOS. Эта команда - основное средство взаимодействия программ с операционной системой. В нашем примере вызывается функция DOS номер 9 - вывести строку на экран. Эта функция выводит строку от начала, адрес которого задается в регистрах DS:DX, до первого встреченного символа \$. При запуске COM-файла регистр DS автоматически загружается сегментным адресом программы, а регистр DX был подготовлен предыдущей командой.

Команда RET используется обычно для возвращения из процедуры. DOS вызывает COM-программы так, что команда RET корректно завершает программу.

Следующая строка программы HELLO-1.ASM определяет строку данных, содержащую текст Hello World!, управляющий символ ASCII возврат каретки с кодом 0Dh, управляющий символ ASCII перевод строки с кодом 0Ah и символ \$, завершающий строку. Эти два управляющих символа переводят курсор на первую позицию следующей строки.

Директива END завершает программу, одновременно указывая, с какой метки должно начинаться ее выполнение.

Программа типа EXE

EXE-программы немного сложнее в исполнении, но для них отсутствует ограничение размера в 64 Кб, так что все достаточно большие программы используют именно этот формат. Ассемблер позволяет уместить и в 64 Кб весьма сложные и большие алгоритмы, а все данные хранить в отдельных файлах, но ограничение

размера все равно очень серьезно, и даже чисто ассемблерные программы могут с ним сталкиваться.

```
;hello-2 asm
;Выводит на экран сообщение "Hello World!" и завершается.
.model small                ;Модель памяти, используемая для EXE.
.stack 100h                 ;Сегмент стека размером в 256 байт,
.code
start: mov ax,DGROUP        ;Сегментный адрес строки message
      mov ds,ax             ;помещается в DS.
      mov dx,offset message
      mov ah,9
      int 21h               ;Функция DOS "вывод строки".
      mov ax,4C00h
      int 21h               ;Функция DOS "завершить программу".
      .data
message db "Hello World!",0Dh,0Ah,'$'
end start
```

В этом примере определяются три сегмента - **сегмент стека** директивой `.STACK` размером в 256 байт; **сегмент кода**, начинающийся с директивы `.CODE`; и **сегмент данных**, начинающийся с `.DATA`. При запуске EXE-программы регистр DS уже не содержит адреса сегмента со строкой message (он указывает на сегмент с блоком данных), а для вызова используемой функции DOS этот регистр должен иметь сегментный адрес строки. Команда `MOV AX,DGROUP` загружает в AX сегментный адрес группы сегментов данных DGROUP, а `MOV DS,AX` копирует его в DS. Вместо DGROUP можно использовать предопределенную метку `@data`, но единственная модель памяти, в которой группа сегментов данных называется иначе, - FLAT (ей мы пользоваться не будем). Программы типа EXE должны завершаться системным вызовом DOS 4Ch: в регистр AH помещается значение 4Ch, в регистр AL - код возврата (в данном примере код возврата 0 и регистры AH и AL загружаются одной командой `MOV AX,4C00h`), после чего вызывается прерывание 21h.

Компиляция hello-2.asm.

tasm hello-2 asm ,,,

tlink /x hello-2 obj

Размер получаемого файла hello-2.exe - 559 байт.

Задание на исследование.

1. Реализовать все этапы создания программы типа *.com (Hello-1.asm).
2. Реализовать все этапы создания программы типа *.exe (Hello-1.asm).

В выводах пояснить различия между типами программ.

Все этапы работы продемонстрировать с помощью скриншотов.

Лабораторная работа №2

Тема: Средства ввода языка ассемблер. Ввод с использованием средств DOS.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 131-151.

1. Ввод с клавиатуры.

Таблица 1. - Функции DOS используемые для ввода с клавиатуры
(стандартного устройства ввода STDIN)

Номер функции	Действие	Вход	Выход
01h	Считать символ из STDIN с эхом, ожиданием и проверкой на Ctrl-Break	АН = 01h	AL = ASCII-код
08h	Считать символ из STDIN без эха, с ожиданием и проверкой на Ctrl-Break	АН = 08h	AL = код символа
07h	Считать символ из STDIN без эха, с ожиданием и без проверки на Ctrl-Break	АН = 07h	AL = код символа
06h	Считать символ из STDIN без эха, без ожидания и без проверки на Ctrl-Break	АН = 06h DL = 0FFh	ZF = 1, если клавиша не нажата, AL = 00 ZF = 0, если клавиша нажата, AL = код символа

Для работы с клавиатурой, кроме перечисленных, могут потребоваться и некоторые служебные функций DOS.

Таблица 2. – Служебные функции DOS.

Номер функции	Действие	Вход	Выход
0Bh	Проверить состояние клавиатуры	АН = 0Bh	AL = 0, не была нажата клавиша AL = 0FFh, была нажата клавиша
0Ch	Очистить буфер и считать символ	АН = 0Ch AL = Номер функции DOS (01,06, 07, 08, 0Ah)	Зависит от вызванной функции

Задание на исследование.

1. Используя в качестве основы программу, приведённую в **приложении 2**, составить программы для ввода с клавиатуры, использующие функции **DOS** приведенные в таблицах 1 и 2.

В выводах пояснить различия между применяемыми функциями.

2. Используя в качестве основы программу, приведённую в **приложении 3**, составить программы для ввода с клавиатуры, использующие функции **DOS** приведенные в таблицах 1 и 2.

В выводах пояснить различия между применяемыми функциями.

Все этапы работы продемонстрировать с помощью скриншотов.

Приложение 1. Таблица ASCII кодов.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺	☹	♥	♦	♣	♠	●	◻	○	◼	♂	♀	♪	♫	☼
10	▶	◀	↑	!!	¶	§	-	↓	↑	↓	→	←	↶	↷	▲	▼
20		!	"	#	\$	%	&	'	(	)	*	+	,	—	.	/
30	Ø	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	◊

Номера строк соответствуют первой цифре в шестнадцатеричном коде символа, номера столбцов – второй. (например, A – 41h).

Приложение 2. Шаблон программы *.com.

;Prog-1.asm

;Считывает символ с клавиатуры выводит его на экран и завершается.

.model tiny ;Модель памяти, используемая для COM.

.code ;Начало сегмента кода.

org 100h ;Начальное значение счетчика - 100h.

start: mov ah,08h ;Номер функции DOS - в АН.

int 21h ;Вызов системной функции DOS.

mov ah,02h ;Номер функции DOS - в АН.

mov dl,al

int 21h ;Вызов системной функции DOS.

ret ;Завершение COM-программы.

end start ;Конец программы.

Приложение 3. Шаблон программы *.exe.

;Prog-2.asm

;Считывает символ с клавиатуры выводит его на экран и завершается.

.model small ;Модель памяти, используемая для EXE.

.stack 100h ;Сегмент стека размером в 256 байт,

.code

start: mov ah,08h ;Номер функции DOS - в АН

int 21h ;Вызов системной функции DOS

```

mov    ah,02h                ;Номер функции DOS - в AH.
mov    dl,al
int    21h                  ;Вызов системной функции DOS.
mov    ax,4C00h
int    21h                  ;Функция DOS "завершить программу".
end    start

```

Лабораторная работа №3

Тема: Средства ввода языка ассемблер. Ввод с использованием средств BIOS.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 131-151.

1. Ввод с клавиатуры.

Таблица 1. - Функции BIOS используемые для ввода с клавиатуры
(стандартного устройства ввода STDIN)

Номер функции	Действие	Вход	Выход
00h, 10h, 20h	Чтение символа с ожиданием	AH = =00h (83/84-key), =10h (101/102-key), =20h (122-key)	AL = ASCII-код символа, AH = скан-код нажатой клавиши
01h, 11h, 21h	Проверка символа	AH = =01h (83/84-key), =11h (101/102-key), =21h (122-key)	ZF = 1, если буфер пуст ZF = 0, если в буфере присутствует символ, тогда AL = ASCII-код символа, AH = скан-код нажатой клавиши.
05h	Поместить символ в буфер клавиатуры	AH = 05h CH = скан-код CL = ASCII-код	AL – 00h, если операция выполнена успешно; AL = 01h, если буфер клавиатуры переполнен.
02h, 12h, 22h	Считать состояние клавиатуры	AH = =02h (83/84-key), =12h (101/102-key), =22h (122-key)	AL = байт состояния клавиатуры 1 ¹ ; AH = байт состояния клавиатуры 2 (только для

Номер функции	Действие	Вход	Выход
			функций 12h и 22h) ² .

¹Байт состояния клавиатуры 1 (этот байт всегда расположен в памяти по адресу 0000h:0417h или 0040h:0017h):

бит 7: Ins включена
бит 6: CapsLock включена
бит 5: NumLock включена
бит 4: ScrollLock включена
бит 3: Alt нажата (любая Alt для функции 02h, часто только левая Alt для 12h/22h)
бит 2: Ctrl нажата (любая Ctrl)
бит 1: Левая Shift нажата
бит 0: Правая Shift нажата

²Байт состояния клавиатуры 2 (этот байт всегда расположен в памяти по адресу 0000h:0418h или 0040h:0018h):

бит 7: SysRq нажата
бит 6: CapsLock нажата
бит 5: NumLock нажата
бит 4: ScrollLock нажата
бит 3: Правая Alt нажата
бит 2: Правая Ctrl нажата
бит 1: Левая Alt нажата
бит 0: Левая Ctrl нажата

Оба байта постоянно располагаются в памяти, так что вместо вызова прерывания часто удобнее просто считывать значения напрямую. Более того, в эти байты можно записывать новые значения, и BIOS соответствующим образом изменит состояние клавиатуры.

Для вызова системных функций BIOS, приведенных в таблице 1 используется команда INT 16h (вызов прерывания с номером 16h).

Задание на исследование.

1. Используя в качестве основы программу, приведённую в **приложении 2**, составить программы для ввода с клавиатуры, использующие функции BIOS приведенные в таблице 1.

В выводах пояснить различия между применяемыми функциями.

2. Используя в качестве основы программу, приведённую в **приложении 3**, составить программы для ввода с клавиатуры, использующие функции BIOS приведенные в таблице 1.

В выводах пояснить различия между применяемыми функциями.

Все этапы работы продемонстрировать с помощью скриншотов.

Приложение 1. Таблица ASCII кодов.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺	☹	♥	♦	♣	♠	●	◻	○	◼	♂	♀	♪	♫	☼
10	▶	◀	↑	!!	¶	§	-	↑	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	(	)	*	+	,	—	.	/
30	Ø	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	␣

Номера строк соответствуют первой цифре в шестнадцатеричном коде символа, номера столбцов – второй. (например, A – 41h).

Приложение 2. Шаблон программы *.com.

```

;Prog-1.asm
;Считывает символ с клавиатуры выводит его на экран и завершается.
.model tiny                                ;Модель памяти, используемая для COM.
.code                                     ;Начало сегмента кода.
org 100h                                ;Начальное значение счетчика - 100h.
start:  mov ah,08h                      ;Номер функции DOS - в АН.
        int 21h                        ;Вызов системной функции DOS.
        mov ah,02h                    ;Номер функции DOS - в АН.
        mov dl,al
        int 21h                        ;Вызов системной функции DOS.
        ret                            ;Завершение COM-программы.
end start                                ;Конец программы.

```

Приложение 3. Шаблон программы *.exe.

```

;Prog-2.asm
;Считывает символ с клавиатуры выводит его на экран и завершается.
.model small                              ;Модель памяти, используемая для EXE.
.stack 100h                              ;Сегмент стека размером в 256 байт,
.code
start:  mov ah,08h                      ;Номер функции DOS - в АН
        int 21h                        ;Вызов системной функции DOS
        mov ah,02h                    ;Номер функции DOS - в АН.
        mov dl,al
        int 21h                        ;Вызов системной функции DOS.
        mov ax,4C00h
        int 21h                        ;Функция DOS "завершить программу".
end start

```

Лабораторная работа №4

Тема: Средства вывода языка ассемблер. Вывод с использованием средств DOS и BIOS.

1. Вывод на экран средствами DOS.
2. Вывод на экран средствами BIOS.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 131-151.

1. Вывод на экран средствами DOS.

Таблица 1. Функции DOS используемые для вывода данных на экран

Номер функции	Действие	Вход	Выход
02h	Записать символ в STDOUT с проверкой на Ctrl-Break	AH = 02h DL = ASCII-код символа	AL = код последнего записанного символа
06h	Записать символ в STDOUT без проверки на Ctrl-Break	AH = 06h DL = ASCII-код символа (кроме 0FFh)	AL = код записанного символа (копия DL)
09h	Записать строку в STDOUT с проверкой на Ctrl-Break	AH = 09h DS:DX = адрес строки, заканчивающейся символом \$ (24h)	AL = 24h (код последнего символа)

Для вызова системных функций DOS, приведенных в таблице 1 используется команда **INT 21h** (вызов прерывания с номером 21h), эта команда обычно следует непосредственно за командой загрузки в регистр AH номера функции DOS, например:

```
mov ah, 9          ;Номер функции DOS - в AH.  
int 21h           ;Вызов системной функции DOS.
```

2. Вывод на экран средствами BIOS

Таблица 2. - Функции BIOS используемые для вывода данных на экран

Номер функции	Действие	Вход	Выход
00h	Установить видеорежим	AH - 00h AL - номер режима в младших 7 битах ³	-
4F02h	Установить видеорежим SuperVGA	AX = 4F02h BX - номер режима в младших 13 битах	-
02h	Установить положение курсора	AH = 02h BH = номер страницы DH = строка DL = столбец	-

Номер функции	Действие	Вход	Выход
03h	Считать положение и размер курсора	АН = 03h ВН - номер страницы	ДН, DL = строка и столбец текущей позиции курсора; СН, CL = первая и последняя строки курсора
08h	Считать символ и атрибут символа ⁴ в текущей позиции курсора	АН = 08h ВН = номер страницы	АН = атрибут символа; AL = ASCII-код символа
09h	Вывести символ с заданным атрибутом на экран	АН = 09h ВН = номер страницы AL = ASCII-код символа BL = атрибут символа СХ = число повторений символа	-
0Ah	Вывести символ с текущим атрибутом на экран	АН = 0Ah ВН = номер страницы AL = ASCII-код символа СХ = число повторений символа	-
0Eh	Вывести символ в режиме телетайпа	АН = 0Eh ВН = номер страницы AL = ASCII-код символа	-
13h	Вывести строку символов с заданными атрибутами	АН - 13h AL = режим вывода ⁵ СХ = длина строки (только число символов) BL = атрибут, если строка содержит только символы ДН – номер строки, DL – номер столбца, начиная с которых будет выводиться строка ВР = адрес начала строки в памяти	-

³Вызов этой функции приводит к тому, что экран переводится в выбранный режим. Если старший бит AL не установлен в 1, экран очищается. Номера текстовых режимов:

0 и 1 - 16-цветные режимы 40x25 (с 25 строками по 40 символов в строке),

2 и 3 - 16-цветные режимы 80x25,

7 - монохромный режим 80x25.

⁴Атрибут символа

-бит 7: символ мигает (по умолчанию) или фон яркого цвета (если его действие было переопределено видеофункцией 10h);

-биты 6-4: цвет фона;

-бит 3: символ яркого цвета (по умолчанию) или фон мигает (если его действие было переопределено видеофункцией 11h);

-биты 2-0: цвет символа.

Цвета кодируются в битах, как показано в таблице 3.

⁵бит 0: переместить курсор в конец строки после вывода;

бит 1: строка содержит не только символы, но и атрибуты, так что каждый символ описывается двумя байтами: ASCII-код и атрибут;

биты 2-7: зарезервированы.

Для вызова системных функций BIOS, приведенных в таблице 2 используется команда INT 10h (вызов прерывания с номером 10h).

Таблица 3. Атрибуты символов

Атрибут	Обычный цвет	Яркий цвет
0000b	Черный	Темно-серый
001b	Синий	Светло-синий
010b	Зеленый	Светло-зеленый
011b	Голубой	Светло-голубой
100b	Красный	Светло-красный
101b	Пурпурный	Светло-пурпурный
110b	Коричневый	Желтый
111b	Светло-серый	Белый

Задание на исследование.

1. Используя в качестве основы программу, приведённую в **приложении 2**, составить программы ввода-вывода использующие функции **DOS** приведенные в таблице 1.

В выводах пояснить различия между применяемыми функциями.

2. Используя в качестве основы программу, приведённую в **приложении 3**, составить программы ввода-вывода использующие функции **DOS** приведенные в таблице 1.

В выводах пояснить различия между применяемыми функциями.

3. Используя в качестве основы программу, приведённую в **приложении 2**, составить программы ввода-вывода использующие функции **BIOS** приведенные в таблице 2.

В выводах пояснить различия между применяемыми функциями.

4. Используя в качестве основы программу, приведённую в **приложении 3**, составить программы ввода-вывода использующие функции **BIOS** приведенные в таблице 2.

В выводах пояснить различия между применяемыми функциями.

Все этапы работы продемонстрировать с помощью скриншотов.

Приложение 1. Таблица ASCII кодов.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺	☹	♥	♦	♣	♠	●	◻	○	◼	♂	♀	♪	♫	☼
10	▶	◀	↕	!!	¶	§	-	↑	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	(	)	*	+	,	—	.	/
30	Ø	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△

Номера строк соответствуют первой цифре в шестнадцатеричном коде символа, номера столбцов – второй. (например, A – 41h).

Приложение 2. Шаблон программы *.com.

```

;Prog-1.asm
;Считывает символ с клавиатуры выводит его на экран и завершается.
.model tiny                ;Модель памяти, используемая для COM.
.code                      ;Начало сегмента кода.
org 100h                  ;Начальное значение счетчика - 100h.
start:  mov ah,08h        ;Номер функции DOS - в AH.
        int 21h           ;Вызов системной функции DOS.
        mov ah,02h        ;Номер функции DOS - в AH.
        mov dl,al
        int 21h           ;Вызов системной функции DOS.
        ret               ;Завершение COM-программы.
end start                 ;Конец программы.

```

Приложение 3. Шаблон программы *.exe.

```

;Prog-2.asm
;Считывает символ с клавиатуры выводит его на экран и завершается.
.model small              ;Модель памяти, используемая для EXE.
.stack 100h              ;Сегмент стека размером в 256 байт,
.code
start:  mov  ah,08h        ;Номер функции DOS - в AH
        int  21h          ;Вызов системной функции DOS
        mov  ah,02h        ;Номер функции DOS - в AH.
        mov  dl,al
        int  21h          ;Вызов системной функции DOS.
        mov  ax,4C00h
        int  21h          ;Функция DOS "завершить программу".
end     start

```

Приложение 4. Пример использования функции BIOS 13h

```

.model tiny
.code
ORG 100h                ;начало COM-файла
start:

```

```

MOV     AH, 13h          ;Номер функции 13h
MOV     AL, 1            ;Перевести курсор в конец строки
MOV     CX, 15           ;Длина строки
MOV     BL, 00011110b    ;Жёлтый текст на синем фоне
MOV     DL, 5            ;Координата X
MOV     DH, 2            ;Координата Y
MOV     BP, offset stroka ;Адрес строки записываем в DX
INT     10h
RET                                ;завершение COM-файла

stroka  DB 'Hello, World!!!!' ;Строка для вывода
END     start

```

Лабораторная работа №5

Тема: Средства вывода языка ассемблер.
Вывод с использованием средств BIOS.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 131-151.

Таблица 1 - Функции BIOS используемые для вывода данных на экран

Номер функции	Действие	Вход	Выход
00h	Установить видеорежим	AH - 00h AL - номер режима в младших 7 битах ³	-
4F02h	Установить видеорежим SuperVGA	AX = 4F02h BX - номер режима в младших 13 битах	-
02h	Установить положение курсора	AH = 02h BH = номер страницы DH = строка DL = столбец	-
03h	Считать положение и размер курсора	AH = 03h BH - номер страницы	DH, DL = строка и столбец текущей позиции курсора; CH, CL = первая и последняя строки курсора
08h	Считать символ и атрибут символа ⁴ в текущей позиции курсора	AH = 08h BH = номер страницы	AH = атрибут символа; AL = ASCII-код символа
09h	Вывести символ с заданным атрибутом на экран	AH = 09h BH = номер страницы	-

Номер функции	Действие	Вход	Выход
		AL = ASCII-код символа BL = атрибут символа CX = число повторений символа	
0Ah	Вывести символ с текущим атрибутом на экран	AH = 0Ah BH = номер страницы AL = ASCII-код символа CX = число повторений символа	-
0Eh	Вывести символ в режиме телетайпа	AH = 0Eh BH = номер страницы AL = ASCII-код символа	-
13h	Вывести строку символов с заданными атрибутами	AH - 13h AL = режим вывода ⁵ CX = длина строки (только число символов) BL = атрибут, если строка содержит только символы DH – номер строки, DL – номер столбца, начиная с которых будет выводиться строка BP = адрес начала строки в памяти	-

³Вызов этой функции приводит к тому, что экран переводится в выбранный режим. Если старший бит AL не установлен в 1, экран очищается. Номера текстовых режимов:

0 и 1 - 16-цветные режимы 40x25 (с 25 строками по 40 символов в строке),
2 и 3 - 16-цветные режимы 80x25,
7 - монохромный режим 80x25.

⁴Атрибут символа

-бит 7: символ мигает (по умолчанию) или фон яркого цвета (если его действие было переопределено видеofункцией 10h);

-биты 6-4: цвет фона;

-бит 3: символ яркого цвета (по умолчанию) или фон мигает (если его действие было переопределено видеofункцией 11h);

-биты 2-0: цвет символа.

Цвета кодируются в битах, как показано в таблице 3.

⁵бит 0: переместить курсор в конец строки после вывода;

бит 1: строка содержит не только символы, но и атрибуты, так что каждый символ описывается двумя байтами: ASCII-код и атрибут;
биты 2-7: зарезервированы.

Для вызова системных функций BIOS, приведенных в таблице 1 используется команда INT 10h (вызов прерывания с номером 10h).

Таблица 2. Атрибуты символов

Атрибут	Обычный цвет	Яркий цвет
0000b	Черный	Темно-серый
001b	Синий	Светло-синий
010b	Зеленый	Светло-зеленый
011b	Голубой	Светло-голубой
100b	Красный	Светло-красный
101b	Пурпурный	Светло-пурпурный
110b	Коричневый	Желтый
111b	Светло-серый	Белый

Задание на занятие.

1. Используя в качестве основы программы, приведённые в **Приложении 2** и **Приложении 4**, составить программы (*.com) вывода на экран использующие функции BIOS приведенные в таблице 1.

В выводах пояснить различия между применяемыми функциями.

2. Используя в качестве основы программу, приведённую в **Приложении 3** и **Приложении 4**, составить программы (*.exe) вывода на экран использующие функции BIOS приведенные в таблице 1.

В выводах пояснить различия между применяемыми функциями.

Все этапы работы продемонстрировать с помощью скриншотов.

Приложение 1. Таблица ASCII кодов.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺	☹	♥	♦	♣	♠	●	◻	○	◼	♂	♀	♪	♫	☼
10	▶	◀	↑	!!	¶	§	-	↑	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	(	)	*	+	,	—	.	/
30	Ø	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	␣

Номера строк соответствуют первой цифре в шестнадцатеричном коде символа, номера столбцов – второй. (например, A – 41h).

Приложение 2. Шаблон программы *.com.

```

;Prog-1.asm
;Считывает символ с клавиатуры выводит его на экран и завершается.
.model tiny                ;Модель памяти, используемая для COM.
.code                      ;Начало сегмента кода.
org 100h                   ;Начальное значение счетчика - 100h.
start:                     ;Номер функции DOS - в АН.
    mov ah,08h             ;Вызов системной функции DOS.
    int 21h                ;Номер функции DOS - в АН.
    mov ah,02h             ;Вызов системной функции DOS.
    mov dl,al              ;Завершение COM-программы.
    int 21h                ;Конец программы.
    ret
end start

```

Приложение 3. Шаблон программы *.exe.

```

;Prog-2.asm
;Считывает символ с клавиатуры выводит его на экран и завершается.
.model small                ;Модель памяти, используемая для EXE.
.stack 100h                 ;Сегмент стека размером в 256 байт,
.code
start: mov ah,08h           ;Номер функции DOS - в АН
    int 21h                ;Вызов системной функции DOS
    mov ah,02h             ;Номер функции DOS - в АН.
    mov dl,al              ;Вызов системной функции DOS.
    int 21h                ;Функция DOS "завершить программу".
    mov ax,4C00h
    int 21h
end start

```

Приложение 4. Пример использования функции BIOS 13h

```

.model tiny
.code
ORG 100h ;начало COM-файла
start:
MOV AH, 13h ;Номер функции 13h
MOV AL, 1 ;Перевести курсор в конец строки
MOV CX, 15 ;Длина строки
MOV BL, 00011110b ;Жёлтый текст на синем фоне
MOV DL, 5 ;Координата X
MOV DH, 2 ;Координата Y
MOV BP, offset stroka ;Адрес строки записываем в DX
INT 10h
RET ;завершение COM-файла

stroka DB 'Hello, World!!!!' ;Строка для вывода
END start

```

Лабораторная работа №6

Тема: Целочисленные арифметические команды. Двоичная арифметика.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 34-41.

Команды двоичной арифметики.

Таблица 1 - Непривилегированные команды двоичной арифметики.

Команда	Назначение	Синтаксис	Операнды
ADD	сложение	ADD приемник, источник	Приемник - регистр или переменная, источник - число, регистр или переменная, (нельзя использовать переменную одновременно и для источника, и для приемника).
ADC	сложение с переносом	ADC приемник, источник	См. ADD .
INC	инкремент	INC приемник	Приемник - регистр или переменная.
SUB	вычитание	SUB приемник, источник	Приемник - регистр или переменная, источник - число, регистр или переменная, (нельзя использовать переменную одновременно и для источника, и для приемника).

Команда	Назначение	Синтаксис	Операнды
SBB	вычитание с займом	SBB приемник, источник	См. SUB .
DEC	декремент	INC приемник	Приемник - регистр или переменная.
MUL	умножение чисел без знака	MUL источник	Источник - регистр или переменная, приемник неявно в AL, AX, EAX.
IMUL	умножение чисел со знаком	IMUL источник IMUL приемник, источник	IMUL источник: источник - регистр или переменная (приемник неявно в AL, AX или EAX). IMUL приемник, источник: источник - число, регистр или переменная, приемник - регистр.
DIV	целочисленное деление без знака	DIV источник	Приемник - AL, AX или EAX делится на источник - регистр или переменная.
IDIV	целочисленное деление со знаком	IDIV источник	См. DIV .

Задания для самостоятельного выполнения.

Таблица 2 - Варианты заданий

№	Задание	Тип исполнительного модуля
	Найти сумму и разность двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.com
	Найти произведение и сумму двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.exe
	Найти разность и произведение двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.com
	Найти произведение и частное от деления двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.exe
	Найти частное от деления двух чисел и их сумму - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.exe
	Найти сумму и произведение двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.com
	Найти разность и сумму двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.exe
	Найти частное от деления двух чисел и их сумму	.com

№	Задание	Тип исполнительного модуля
	- программа с вводом с клавиатуры, без корректировки результата	
	Найти разность и частное от деления двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.exe
	Найти разность и произведение двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.com
	Найти частное от деления двух чисел и их сумму - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.exe
	Найти сумму и разность двух чисел - программа без ввода с клавиатуры и корректировки результата - программа с вводом с клавиатуры, без корректировки результата	.com

Лабораторная работа №7

Тема: Команды передачи управления.

Учебные вопросы.

1. Команды безусловной передачи управления.
2. Команды условной передачи управления.

Литература:

С.В. Зубков Assembler для DOS, Windows и UNIX. -М.: ДМК, 2000. с. 47-54.
Assembler/ В.Юров. -СПб.: Питер 2001. с. 202-228.

Время: 6 часов.

1. Описание метки.

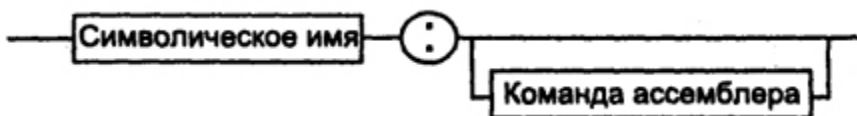


Рисунок 1 - Синтаксис описания метки оператором (:)

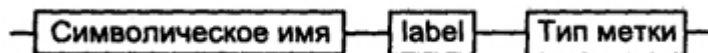


Рисунок 2 - Синтаксис директивы label

2. Описание процедуры.

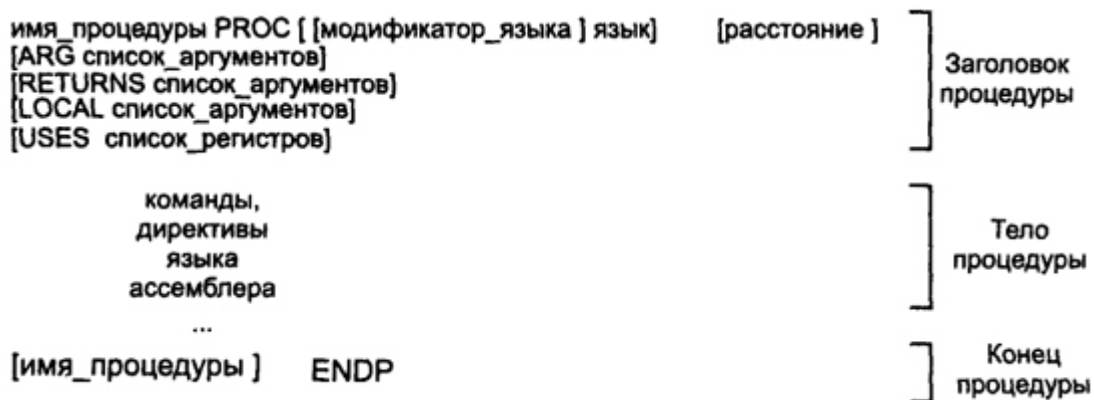


Рисунок 3 - Синтаксис описания процедуры в программе

3. Условные переходы.

Таблица 1 - Значение аббревиатур в названии команды Jcc

Мнемоническое обозначение	Английский	Русский	Тип операндов
Ee	equal	Равно	Любые
Np	not	Не	Любые
Gg	greater	Больше	Числа со знаком
Ll	less	Меньше	Числа со знаком
Aa	above	Выше, в смысле "больше"	Числа без знака
Bb	below	Ниже, в смысле "меньше"	Числа без знака

Таблица 2 - Перечень команд условного перехода для команды стр операнд_1,операнд_2

Типы операндов	Мнемокод команды условного перехода	Критерий условного перехода	Значения флагов для осуществления перехода
Любые	je	операнд_1 = операнд_2	zf = 1
Любые	jne	операнд_1 <> операнд_2	zf = 0
Со знаком	jl/jnge	операнд_1 < операнд_2	sf <> of
Со знаком	jle/jng	операнд_1 <= операнд_2	sf <> of or zf = 1
Со знаком	jg/jnle	операнд_1 > операнд_2	sf = of and zf = 0
Со знаком	jge/jnl	операнд_1 >= операнд_2	sf = of
Без знака	jb/jnae	операнд_1 < операнд_2	cf = 1
Без знака	jbe/jna	операнд_1 <= операнд_2	cf = 1 or zf=1
Без знака	ja/jnbe	операнд_1 > операнд_2	cf = 0 and zf = 0
Без знака	jae/jnb	операнд_1 >= операнд_2	cf = 0

Таблица 3 - Команды условного перехода и флаги

Название флага	Номер бита в eflags/flags	Команда условного перехода	Значение флага для осуществления перехода
Флаг переноса cf	1	jc	cf=1
Флаг четности pf	2	jP	pf=1
Флаг нуля zf	6	jz	zf=1
Флаг знака sf	7	js	sf= 1
Флаг переполнения of	11	jo	of=1
Флаг переноса cf	1	jnc	cf=0
Флаг четности pf	2	jnp	pf=0
Флаг нуля zf	6	jnz	zf=0
Флаг знака sf	7	jns	sf=0
Флаг переполнения of	11	jno	of=0

Задания для самостоятельного выполнения.

Таблица 3. - Варианты заданий

№	Задание	Тип исполнительного модуля
	Найти максимум из трех чисел	.exe
	Найти минимум из трех чисел	.com
	Найти сумму нулей в двоичной четырехзначной последовательности	.exe
	Найти сумму единиц в двоичной четырехзначной последовательности	.com
	Найти максимум из трех чисел	.com
	Найти минимум из трех чисел	.exe
	Для четырехзначного числа найти сумму элементов <1	.com
	Найти сумму нулей в двоичной четырехзначной последовательности	.com
	Найти сумму единиц в двоичной четырехзначной последовательности	.exe
	Для четырехзначного числа найти сумму элементов <1	.exe
	Инвертировать содержимое двоичной четырехразрядной последовательности	.com
	Инвертировать содержимое двоичной четырехразрядной последовательности	.exe
	Сортировать двоичную четырехзначную последовательность в порядке 0-1	.com
	Сортировать двоичную четырехзначную последовательность в порядке 1-0	.exe

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕ- ЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для организации и проведения самостоятельной работы
студентов по дисциплине «Системное программирование»
направления подготовки 09.03.01 Информатика и вычислительная техника,
направленность (профиль) «Программное обеспечение средств вычислительной техники и
автоматизированных систем»

Ставрополь, 2026

Методические указания по организации самостоятельной работы студентов составлено в соответствии с требованиями программы дисциплины «Системное программирование» для бакалавров направления подготовки 09.03.01 Информатика и вычислительная техника. Учебно-методическое пособие посвящено планируемой учебной работе студентов, выполняемой во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия, при частичном, непосредственном участии преподавателя.

СОДЕРЖАНИЕ

1. Самостоятельная работа как важнейшая форма учебного процесса
2. Цели и основные задачи СРС
3. Виды самостоятельной работы
4. Организация СРС
5. Общие рекомендации по организации самостоятельной работы
6. Самостоятельная работа студента - необходимое звено становления исследователя
7. Методические рекомендации для студентов по отдельным формам самостоятельной работы
8. Учебно-методическое и информационное обеспечение дисциплины

1. Самостоятельная работа как важнейшая форма учебного процесса

Самостоятельная работа – это планируемая учебная, учебно- исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.

Концепцией модернизации российского образования определены основные задачи профессионального образования - "подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности".

Решение этих задач невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание творческой активности и инициативы.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в лабораторных или практических занятиях, в выполнении контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Формы самостоятельной работы студентов разнообразны. По данной дисциплине «Системное программирование» предусмотрены следующие:

Использование материала учебно-методического комплекса дисциплины

На первом этапе необходимо ознакомиться с обязательным минимумом содержания основной образовательной программы по изучаемой дисциплине. Далее необходимо изучить рабочую программу дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций лабораторных/практических занятий, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, представленные в учебной программе по дисциплине «Системное программирование».

Работа с литературой

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить

представленные темы по рекомендуемым источникам информации, которые представлены в учебной программе.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

2. Цели и основные задачи СРС

Ведущая цель организации и осуществления СРС должна совпадать с целью обучения студента – подготовкой бакалавра с высшим образованием. При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Виды самостоятельной работы

В образовательном процессе высшего профессионального образовательного учреждения выделяется два вида самостоятельной работы – аудиторная, под руководством преподавателя, и внеаудиторная. Тесная взаимосвязь этих видов работ предусматривает дифференциацию и эффективность результатов ее выполнения и зависит от организации, содержания, логики учебного процесса (межпредметных связей, перспективных знаний и др.):

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Основными видами самостоятельной работы студентов без участия преподавателя по данной дисциплине являются:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);

- подготовка к лабораторным или практическим работам, их оформление.

Основными видами самостоятельной работы студентов с участием преподавателей являются:

- текущие консультации;
- прием и защита лабораторных работ (во время проведения л/р, если они предусмотрены учебным планом);
- выполнение заданий на практических занятиях (если таковые предусмотрены учебным планом дисциплины);
- выполнение курсовой работы или проекта, если они предусмотрены учебным планом.

4. Организация СРС

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Системное программирование», объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Деятельность студентов по формированию и развитию навыков учебной самостоятельной работы

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

Выполняя самостоятельную работу под контролем преподавателя студент должен:

- освоить минимум содержания, выносимый на самостоятельную работу студентов и предложенный преподавателем в соответствии с Федеральными государственными образовательными стандартами высшего образования;
- планировать самостоятельную работу в соответствии с графиком самостоятельной работы, предложенным преподавателем;
- самостоятельную работу студент должен осуществлять в организационных формах, предусмотренных учебным планом и рабочей программой преподавателя;
- выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

студент может:

сверх предложенного преподавателем (при обосновании и согласовании с ним) минимума обязательного содержания, определяемого программой по данной дисциплине:

- самостоятельно определять уровень (глубину) проработки содержания материала;
- предлагать темы и вопросы для самостоятельной проработки;
- в рамках общего графика выполнения самостоятельной работы предлагать обоснованный индивидуальный график выполнения и отчетности по результатам самостоятельной работы;
- предлагать свои варианты организационных форм самостоятельной работы;
- использовать для самостоятельной работы методические пособия, учебные пособия, разработки сверх предложенного преподавателем перечня;
- использовать не только контроль, но и самоконтроль результатов самостоятельной работы в соответствии с методами самоконтроля, предложенными преподавателем или выбранными самостоятельно.

Самостоятельная работа студентов должна оказывать важное влияние на формирование личности будущего профессионала, она планируется студентом самостоятельно. Каждый студент самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием по каждой дисциплине. Он выполняет внеаудиторную работу по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

5. Общие рекомендации по организации самостоятельной работы

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на лабораторных или практических занятиях (в зависимости от учебного плана). Но для успешной учебной деятельности, ее интенсификации, необходимо учитывать следующие субъективные факторы:

1. Знание школьного программного материала, наличие прочной системы знаний, необходимой для усвоения основных вузовских курсов. Это особенно важно для математических и инженерных дисциплин. Необходимо отличать пробелы в знаниях, затрудняющие усвоение нового материала, от малых способностей. Затратив силы на преодоление этих пробелов, студент обеспечит себе нормальную успеваемость и поверит в свои способности.

2. Наличие умений, навыков умственного труда:

- а) умение конспектировать на лекции и при работе с книгой;
- б) владение логическими операциями: сравнение, анализ, синтез, обобщение, определение понятий, правила систематизации и классификации.

3. Специфика познавательных психических процессов: внимание, память, речь, наблюдательность, интеллект и мышление. Слабое развитие каждого из них становится серьезным препятствием в учебе.

4. Хорошая работоспособность, которая обеспечивается нормальным физическим состоянием. Ведь серьезное учение - это большой многосторонний и разнообразный труд. Результат обучения оценивается не количеством сообщаемой информации, а качеством ее усвоения, умением ее использовать и развитием у себя способности к дальнейшему самостоятельному образованию.

5. Соответствие избранной деятельности, профессии индивидуальным способностям. Необходимо выработать у себя умение регулировать свое эмоциональное состояние и устранять обстоятельства, нарушающие деловой настрой, мешающие намеченной работе.

6. Овладение оптимальным стилем работы, обеспечивающим успех в деятельности. Чередование труда и пауз в работе, периоды отдыха, индивидуально обоснованная норма продолжительности сна, предпочтение вечерних или утренних занятий, стрессоустойчивость на экзаменах и особенности подготовки к ним,

7. Уровень требований к себе, определяемый сложившейся самооценкой. Адекватная оценка знаний, достоинств, недостатков - важная составляющая самоорганизации человека, без нее невозможна успешная работа по управлению своим поведением, деятельностью.

Одна из основных особенностей обучения в высшей школе заключается в том, что постоянный внешний контроль заменяется самоконтролем, активная роль в обучении принадлежит уже не столько преподавателю, сколько студенту.

Зная основные методы научной организации умственного труда, можно при наименьших затратах времени, средств и трудовых усилий достичь наилучших результатов.

Эффективность усвоения поступающей информации зависит от работоспособности человека в тот или иной момент его деятельности.

Работоспособность - способность человека к труду с высокой степенью напряженности в течение определенного времени. Различают внутренние и внешние факторы работоспособности.

К внутренним факторам работоспособности относятся интеллектуальные особенности, воля, состояние здоровья.

К внешним:

- организация рабочего места, режим труда и отдыха;
- уровень организации труда - умение получить справку и пользоваться информацией;
- величина умственной нагрузки.

Выдающийся русский физиолог Н. Е. Введенский выделил следующие условия продуктивности умственной деятельности:

- во всякий труд нужно входить постепенно;
- мерность и ритм работы. Разным людям присущ более или менее разный темп работы;
- привычная последовательность и систематичность деятельности;
- правильное чередование труда и отдыха.

Отдых не предполагает обязательного полного бездействия со стороны человека, он может быть достигнут простой переменой дела. В течение дня работоспособность изменяется. Наиболее плодотворным является *утреннее время (с 8 до 14 часов)*, причем максимальная работоспособность приходится на период с 10 до 13 часов, затем *послеобеденное* - (с 16 до 19 часов) и *вечернее* (с 20 до 24 часов). Очень трудный для понимания материал лучше изучать в начале каждого отрезка времени (лучше всего утреннего) после хорошего отдыха. Через 1-1,5 часа нужны перерывы по 10 - 15 мин, через 3 - 4 часа работы отдых должен быть продолжительным - около часа.

Составной частью научной организации умственного труда является овладение техникой умственного труда.

Время, которым располагает студент для выполнения учебного плана, складывается из

двух составляющих: одна из них – это аудиторная работа в вузе по расписанию занятий, другая - внеаудиторная самостоятельная работа. Задания и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль за самостоятельной работой, а также оказывает помощь студентам по правильной организации работы.

Самостоятельные занятия потребуют интенсивного умственного труда, который необходимо не только правильно организовать, но и стимулировать. При этом очень важно уметь поддерживать устойчивое внимание к изучаемому материалу. Выработка внимания требует значительных волевых усилий. Именно поэтому, если студент замечает, что он часто отвлекается во время самостоятельных занятий, ему надо заставить себя сосредоточиться. Подобную процедуру необходимо проделывать постоянно, так как это является тренировкой внимания. Устойчивое внимание появляется тогда, когда человек относится к делу с интересом.

Следует правильно организовать свои занятия по времени: 50 минут - работа, 5-10 минут - перерыв; после 3 часов работы перерыв - 20-25 минут. Иначе нарастающее утомление повлечет неустойчивость внимания. Очень существенным фактором, влияющим на повышение умственной работоспособности, являются систематические занятия физической культурой. Организация активного отдыха предусматривает чередование умственной и физической деятельности, что полностью восстанавливает работоспособность человека.

6. Самостоятельная работа студента - необходимое звено становления исследователя

Прогресс науки и техники, информационных технологий приводит к значительному увеличению научной информации, что предъявляет более высокие требования не только к моральным, нравственным свойствам человека, но и в особенности, постоянно возрастающие требования в области образования – обновление, модернизация общих и профессиональных знаний, умений специалиста.

Всякое образование должно выступать как динамический процесс, присущий человеку и продолжающийся всю его жизнь. Овладение научной мыслью и языком науки является необходимой составляющей в самоорганизации будущего специалиста исследователя. Под этим понимается не столько накопление знаний, сколько овладение научно обоснованными способами их приобретения. В этом, вообще говоря, состоит основная задача вуза.

Специфика вузовского учебного процесса, в организации которого самостоятельной работе студента отводятся все больше места, состоит в том, что он является как будто бы последним и самым адекватным звеном для реализации этой задачи. Поскольку во время учебы в вузе происходит выработка стиля, навыков учебной (познавательной) деятельности, рациональный характер которых будет способствовать постоянному обновлению знаний высококвалифицированного выпускника вуза.

Однако до этого пути существуют определенные трудности, в частности, переход студента от синтетического процесса обучения в средней школе, к аналитическому в высшей. Это связано как с новым содержанием обучения (расширение общего образования и углубление профессиональной подготовки), так и с новыми, неизвестными до сих пор формами: обучения (лекции, семинары, лабораторные занятия и т.д.). Студент получает не только знания, предусмотренные программой и учебными пособиями, но он также должен

познакомиться со способами приобретения знаний так, чтобы суметь оценить, что мы знаем, откуда мы это знаем и как этого знания мы достигли. Ко всему этому приходят через собственную самостоятельную работу.

Это и потому, что самостоятельно приобретенные знания являются более оперативными, они становятся личной собственностью, а также мотивом поведения, развивают интеллектуальные черты, внимание, наблюдательность, критичность, умение оценивать. Роль преподавателя в основном заключается в руководстве накопления знаний (по отношению к первокурсникам), а в последующие годы учебы, на старших курсах, в совместном установлении проблем и заботе о самостоятельных поисках студента, а также контролирования за их деятельностью. Отметим, что нельзя ограничиваться только приобретением знаний, предусмотренных программой изучаемой дисциплины, надо постоянно углублять полученные знания, сосредотачивая их на какой-нибудь узкой определенной области, соответствующей интересам студента. Углубленное изучение всех предметов, предусмотренных программой, на практике является возможным, и хорошая организация работы позволяет экономить время, что создает условия для глубокого, систематического, заинтересованного изучения самостоятельно выбранной студентом темы.

Конечно, все советы, примеры, рекомендации в этой области, даваемые преподавателем, или определенными публикациями, или другими источниками, не гарантируют никакого успеха без проявления собственной активности в этом деле, т. е. они не дают готовых рецептов, а должны способствовать анализу собственной работы, ее целей, организации в соответствии с индивидуальными особенностями. Учитывая личные возможности, существующие условия жизни и работы, навыки, на основе этих рекомендаций, возможно, выработать индивидуально обоснованную совокупность методов, способов, найти свой стиль или усовершенствовать его, чтобы изучив определенный материал, иметь время оценить его значимость, пригодность и возможности его применения, чтобы, в конечном счете, обеспечить успешность своей учебы с будущей профессиональной деятельности

7. Методические рекомендации для студентов по отдельным формам самостоятельной работы

Система вузовского обучения подразумевает значительно большую самостоятельность студентов в планировании и организации своей деятельности.

Работа с литературой

При работе с литературой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой – это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного выяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная

индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента. Различают два вида чтения; первичное и вторичное. *Первичное* - это внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятного слова. Содержание не всегда может быть понятно после первичного чтения.

Задача *вторичного* чтения полное усвоение смысла целого (по счету это чтение может быть и не вторым, а третьим или четвертым).

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того насколько осознанно читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют четыре основные установки в чтении научного текста:

1. информационно-поисковый (задача – найти, выделить искомую информацию);
2. усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения, излагаемые автором, так и всю логику его рассуждений);
3. аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему);
4. творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

С наличием различных установок обращения к научному тексту связано существование и нескольких **видов чтения**:

1. библиографическое – просматривание карточек каталога, рекомендательных списков, сводных списков журналов и статей за год и т.п.;
2. просмотровое – используется для поиска материалов, содержащих нужную информацию, обычно к нему прибегают сразу после работы со списками литературы и каталогами, в результате такого просмотра читатель устанавливает, какие из источников будут использованы в дальнейшей работе;
3. ознакомительное – подразумевает сплошное, достаточно подробное прочтение отобранных статей, глав, отдельных страниц, цель – познакомиться с характером информации, узнать, какие вопросы вынесены автором на рассмотрение, провести сортировку материала;

4. изучающее – предполагает доскональное освоение материала; в ходе такого чтения проявляется доверие читателя к автору, готовность принять изложенную информацию, реализуется установка на предельно полное понимание материала;

5. аналитико-критическое и творческое чтение – два вида чтения близкие между собой тем, что участвуют в решении исследовательских задач. Первый из них предполагает направленный критический анализ, как самой информации, так и способов ее получения и подачи автором; второе – поиск тех суждений, фактов, по которым или в связи с которыми, читатель считает нужным высказать собственные мысли.

Из всех рассмотренных видов чтения основным для студентов является изучающее – именно оно позволяет в работе с учебной литературой накапливать знания в различных областях. Вот почему именно этот вид чтения в рамках учебной деятельности должен быть освоен в первую очередь. Кроме того, при овладении данным видом чтения формируются основные приемы, повышающие эффективность работы с научным текстом.

Основные виды систематизированной записи прочитанного:

1. Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

2. Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

3. Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

4. Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

5. Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.

2. Выделите главное, составьте план.

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и

дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

Лабораторные/Практические занятия (в зависимости от учебного плана)

Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на лабораторных или практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью выполнения заданий лабораторной или практической работы, поставленных задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном выполнении заданий нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение задач или примеров следует излагать по дробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что выполнение каждого учебного задания должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Выполнение заданий данного типа нужно продолжать до приобретения твердых навыков в их решении.

Самопроверка

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих заданий на лабораторных занятиях и самостоятельно студенту рекомендуется, используя лист опорных материалов, воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Консультации

Если в процессе самостоятельной работы над изучением теоретического материала или при выполнении заданий у студента возникают вопросы, разрешить которые самостоятельно не удастся, необходимо обратиться к преподавателю для получения у него разъяснений или указаний. В своих вопросах студент должен четко выразить, в чем он испытывает затруднения, характер этого затруднения. За консультацией следует обращаться и в случае, если возникнут сомнения в правильности ответов на вопросы самопроверки.

Подготовка к зачету по дисциплине «Системное программирование»

Изучение многих общепрофессиональных и специальных дисциплин завершается

зачетом с оценкой. Подготовка к зачету способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания.

Для успешной сдачи зачета с оценкой студенту важно в течение семестра полностью выполнить задания лабораторных или практических работ, подготовить отчет по каждой работе и отчет по результатам самостоятельного изучения тем, согласно учебной программе.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Текущая аттестация студентов проводится преподавателями, ведущими лабораторные или практические занятия по дисциплине в следующих формах: собеседование при защите лабораторной работы или заданий практической работы с предоставлением письменного (или в электронной форме) отчета, в зависимости от предъявляемых требований.

8. Учебно-методическое и информационное обеспечение дисциплины

Основная литература	1. Аблязов Р.З. Программирование на ассемблере на платформе x86-64 [Электронный ресурс]/ Аблязов Р.З. - Электрон. текстовые данные. -М.: ДМК Пресс, 2011. -308 с. - Режим доступа: http://www.iprbookshop.ru/8005 , экземпляров неограниченно.
Дополнительная литература	1. Шелупанов А.А. Информатика. Базовый курс. Часть 1. Общие вопросы информатики и программирование на Ассемблере [Электронный ресурс]: учебник/ Шелупанов А.А., Кирнос В.Н.- Электрон. текстовые данные. -Томск: Томский государственный университет систем управления и радиоэлектроники, В-Спектр, 2007. - 190 с. - Режим доступа: http://www.iprbookshop.ru/14012 , экземпляров неограниченно. 2. Кирнос В.Н. Введение в вычислительную технику. Основы организации ЭВМ и программирование на Ассемблере [Электронный ресурс]: учебное пособие/ Кирнос В.Н. - Электрон. текстовые данные. -Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011. - 172 с. - Режим доступа: http://www.iprbookshop.ru/13921 ., экземпляров неограниченно.
Программное обеспечение	Альт Рабочая станция 10 Альт Рабочая станция К Альт «Сервер» Пакет офисных программ - Р7-Офис