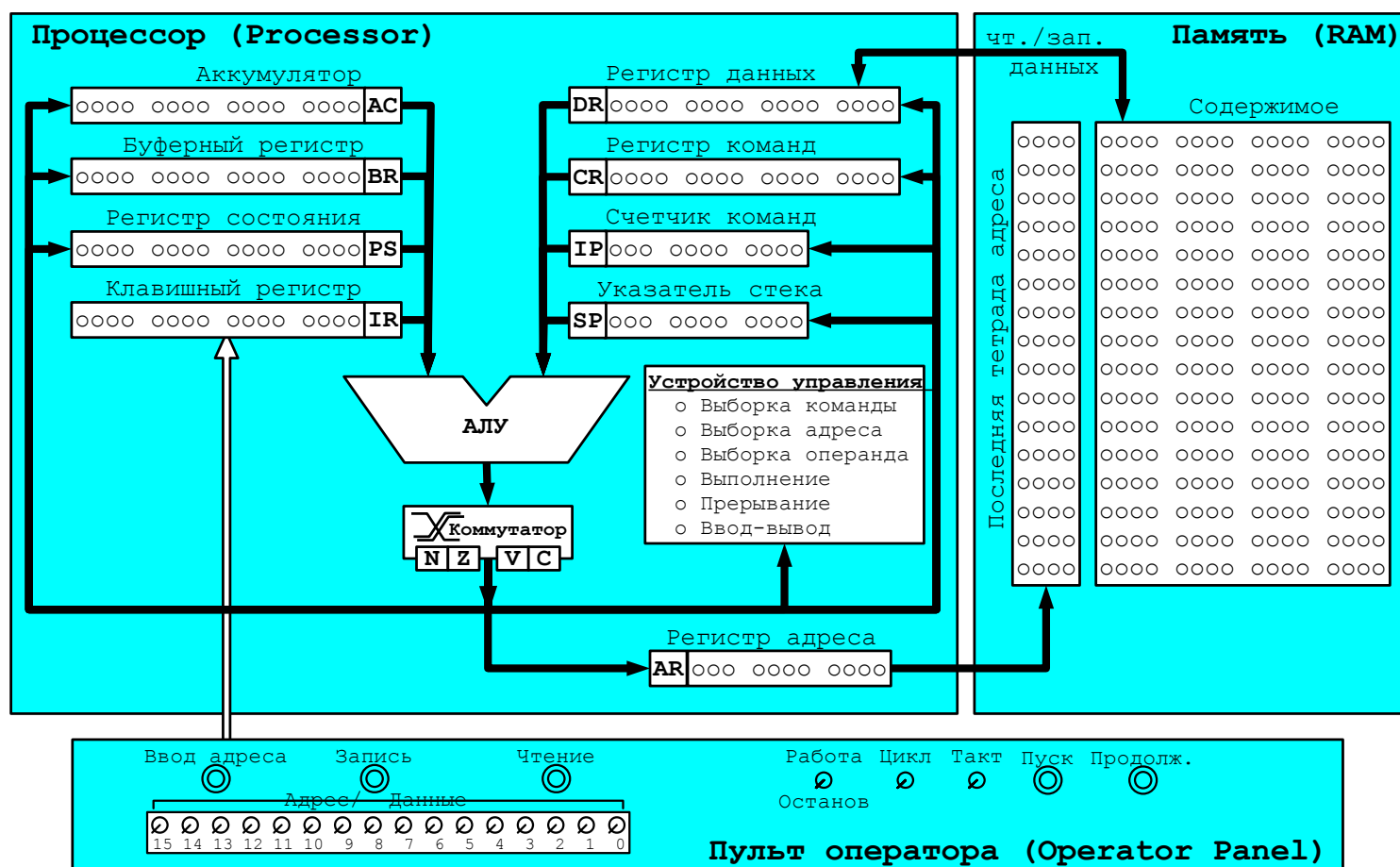


# Методические указания к лабораторным работам по курсу "Основы профессиональной деятельности"





## Содержание

Раздел 1. Знакомство с кафедрой вычислительной техники.....	4
Кафедра вычислительной техники.....	4
Лабораторные работы курса ОПД.....	4
Лабораторная работа №1. <i>Основные команды ОС семейства UNIX</i> .....	5
Рубежный контроль №1.....	6
Раздел 2. Введение в базовую ЭВМ.....	7
Лабораторная работа №2. <i>Исследование работы БЭВМ</i> .....	7
Лабораторная работа №3. <i>Выполнение циклических программ</i> .....	9
Рубежный контроль №2.....	9
Лабораторная работа №4. <i>Выполнение комплекса программ</i> .....	10
Раздел 3. Организация ввода-вывода информации в БЭВМ.....	10
Лабораторная работа №5. <i>Асинхронный обмен данными с ВУ</i> .....	10
Лабораторная работа №6. <i>Обмен данными с ВУ по прерыванию</i> .....	11
Раздел 4. Организация микропрограммного устройства БЭВМ.....	13
Лабораторная работа №7. <i>Синтез команд БЭВМ</i> .....	13
Рубежный контроль №2.....	15
Литература.....	15
Приложение А. Краткий перечень и функциональность команд UNIX.....	16
Приложение Б. Накопление баллов БАРС в течении курса.....	17
Приложение В. Состав, структура и функционирование БЭВМ.....	18
Часть 1. Базовая ЭВМ.....	18
1.1 Назначение базовой ЭВМ.....	18
1.2 Структура базовой ЭВМ.....	18
1.3. Система команд базовой ЭВМ.....	19
1.4 Представление целых чисел в БЭВМ.....	20
1.5 Арифметические операции.....	22
1.6 Сдвиги и логические операции.....	23
1.7 Управление вычислительным процессом.....	23
1.8 Подпрограммы.....	27
1.9 Выполнение машинных команд.....	27
Часть 2. Организация ввода-вывода в базовой ЭВМ.....	31
2.1 Устройства ввода-вывода базовой ЭВМ.....	31
2.2 Команды ввода-вывода.....	32
2.3 Программно-управляемый асинхронный обмен.....	32
2.4 Управляемый по прерыванию программы ввод-вывод.....	34
Часть 3. Микропрограммное устройство управления.....	37
3.1. Микропрограммное управление вентильными схемами.....	37
3.2 Интерпретатор базовой ЭВМ.....	40
Приложение Г. Инструкция по работе с моделью БЭВМ.....	45
Приложение Д. Ассемблер БЭВМ. Краткий справочник.....	46

Настоящее методическое пособие предназначено для практического закрепления материала по дисциплине "Основы вычислительной техники" (ОВТ), преподавание которой организовано по модульному принципу и включает лекции, лабораторные работы, домашние задания и контрольные работы. Курс по основам вычислительной техники и оригинальная модель базовой ЭВМ были разработаны в начале 1980-х годов Кирилловым Владимиром Васильевичем и Приблудой Анатолием Андреевичем. В учебно-методической работе, разработке методических указаний, учебников и моделей базовой ЭВМ, лабораторных работ в разное время принимало участие большое количество сотрудников кафедры вычислительной техники, среди которых особенно хотелось отметить (перечисление в алфавитном порядке) Афанасьева Дмитрия Борисовича, Блохину Елену Николаевну, Гаврилова Антона Валерьевича, Громова Геннадия Юрьевича, Грозову Ирину Владимировну, Дергачева Андрея Михайловича, Клименкова Сергея Викторовича, Лемешева Алексея Сергеевича, Максимова Андрея Николаевича, Майорова Сергея Александровича, Мартянова Николая Васильевича, Перминова Илью Валентиновича, Приблуду Андрея Анатольевича, Приблуду Константина Анатольевича, Щелокова Ивана Викторовича, а также большое количество студентов, аспирантов и выпускников кафедры ВТ.

В 2019 году Афанасьев Д.Б. и Клименков С.В. выступили инициаторами и главными исполнителями проектирования и реализации модели БЭВМ-NG в которую были добавлены необходимые функциональные возможности для обучения студентов, к которым относились стек, режимы адресации, расширенный набор команд, организация ввода-вывода и прерываний. В разработке системы команд и эмулятора также активное участие приняли Медведева Елизавета, Доморацкий Эридан, Щербаков Виктор, Гаврилов Антон Валерьевич и Перминов Илья Валентинович.

В методическом пособии содержится информация, необходимая для успешной сдачи всех лабораторных работ, включая специально разработанную в образовательных целях учебную ЭВМ (базовая ЭВМ), обладающую типичными чертами многих современных ЭВМ. Лабораторная работа №1 предназначена для демонстрации студентам первого курса предмета "Системное программное обеспечение", а также знакомства с лабораториями, где им предстоит обучаться в дальнейшем. Остальные лабораторные работы курса посвящены базовой ЭВМ. С ее помощью студенты исследуют порядок функционирования ЭВМ при выполнении программ различных типов, подходы к организации ввода-вывода информации, принципы микропрограммного управления. В приложениях приведена справочная информация, необходимая при подготовке и защите лабораторных работ.

## **Раздел 1. Знакомство с кафедрой вычислительной техники**

### **Кафедра вычислительной техники**

Кафедра вычислительной техники (ВТ) ведет подготовку бакалавров по направлениям подготовки 09.03.01 - Информатика и Вычислительная техника (профиль подготовки "Вычислительные машины, комплексы, системы и сети") и 09.03.04 - Программная инженерия (профиль подготовки "Разработка программно-информационных систем"). В рамках курса ОВТ ведущие преподаватели направлений подготовки прочитают вводные лекции по своим дисциплинам.

### **Лабораторные работы курса ОПД**

В рамках курса предусмотрено 7 лабораторных работ. Результатом выполнения работы является выполнение всех требований к работе и отчет, который должен включать ряд обязательных составляющих. К ним относятся:

- титульный лист: название университета, кафедры, дисциплины, название и номер лабораторной работы, номер группы и варианта, Ф.И.О. студента, год;
- задание к работе, включая вариант задания;

- порядок выполнения лабораторной работы, дополнительные требования и указания, которые находятся в описании к каждой работе;
- выводы, которые отвечают на вопросы "Что было изучено при выполнении лабораторной работы? Что нового вы узнали? Как можно использовать изученный материал?";
- скрепу, скобу или люверс. Листы должны быть скреплены между собой!

### **Лабораторная работа №1. Основные команды ОС семейства UNIX**

В лабораторных аудиториях кафедры установлено разнообразное вычислительное оборудование под управлением различных операционных систем (ОС). Важное место занимают ОС семейства UNIX, включая различные версии Linux, BSD, Solaris и AIX. Основным способом взаимодействия пользователей и администраторов с такими операционными системами является командный интерфейс с использованием интерпретатора shell.

Цель работы. Знакомство с основным способом взаимодействия с ОС UNIX, командным интерфейсом, а также базовой функциональностью интерпретатора shell. Получение основных сведений о файловой системе и правах доступа к файлам.

#### Задание.

1. Создать приведенное в варианте дерево каталогов и файлов с содержимым. В качестве корня дерева использовать каталог lab0 своего домашнего каталога. Для создания и навигации по дереву использовать команды: `mkdir`, `echo`, `cat`, `touch`, `ls`, `pwd`, `cd`, `more`, `cp`, `rm`, `rmdir`, `mv`.

2. Установить согласно заданию права на файлы и каталоги при помощи команды `chmod`, используя различные способы указания прав.

3. Скопировать часть дерева и создать ссылки внутри дерева согласно заданию при помощи команд `cp` и `ln`, а также команды `cat` и перенаправления ввода-вывода.

4. Используя команды `cat`, `wc`, `ls`, `head`, `tail`, `echo`, `sort`, `grep` выполнить в соответствии с вариантом задания поиск и фильтрацию файлов, каталогов и содержащихся в них данных.

5. Выполнить удаление файлов и каталогов при помощи команд `rm` и `rmdir` согласно варианту задания.

Подготовка к выполнению работы. Изучить справочные страницы по указанным командам. Разобраться с основными принципами организации ввода-вывода с использованием стандартных потоков ввода-вывода (`stdin`, `stdout`, `stderr`, включая перенаправление данных потоков на команды-фильтры и в файлы), типами файлов, правами пользователей на доступ к файлу для операций чтения, записи и исполнения для владельца файла, группы владельца и остальных пользователей системы.

Порядок выполнения работы. Создать указанное в п. 1 задания дерево файлов и каталогов. Обратит внимание на точное соответствие всех атрибутов полученного дерева заданию. Последовательность команд, необходимых для создания дерева, записать в файл по одной команде на строке (командный скрипт) для возможности автоматического повтора команд этого и последующих пунктов. Изменить права на файлы, согласно п.2 задания. Выполнить п.3, включив в скрипт выполняемые команды. В случае недостатка прав для выполнения операции п.3, необходимо изменить командой `chmod` права файлов перед выполнением команды, а после выполнения вернуть их обратно. Выполнить команды вывода содержимого дерева (п.4), сохранить и включить в отчет вывод исполненных команд. Показать полученное дерево преподавателю. Выполнить команды удаления п.5. Включить в отчет последовательность команд удаления с результатом их выполнения.

Содержание отчета по работе. В дополнение к общим обязательным

требованиям, отчет должен содержать:

- Иерархию файлов и каталогов, полученную при помощи команд `ls -lR` из директории `lab0`, после выполнения п.3 задания.
- Задания, команды и результаты их выполнения (если есть), включая сообщения о возникающих ошибках и, для п.3 и п.5, команды, исправляющую ошибки.
- Файл с последовательностью команд по всей лабораторной работе.

Контрольные вопросы:

1. Расскажите про команду `{имя команды}`. Какие она принимает аргументы (их количество, формат, способ задания)? Является ли данная команда фильтром?
2. Стандартные потоки ввода-вывода, назначение. Способы управления стандартными потоками в shell.
3. Стандартные права доступа к файлам и каталогам. Способы задания прав при помощи команды `chmod`. Интерпретация вывода команды `ls -l`.
4. Файлы в ОС UNIX. Типы файлов. Символические и жесткие ссылки.

## Рубежный контроль №1

В рубежном контроле раздела три задания.

Задание 1. Привести последовательность команд, создающих приведенное дерево каталогов и файлов с содержимым, начиная с создания директории `lab0`.

```
/home/s113369/lab0 (каталог)
```

```
+-darumakal (каталог)
```

```
|  +--wurmple (каталог)
```

```
|  +--chinchou (каталог)
```

```
+-houndour1 (файл)
```

Содержимое файлов

`houndour1:`

Способности Howl Smog Roar Bite Odor Sleuth

Beat Up Fire Flamethrower Crunch

Nasty Plot Inferno

Различными способами установить права на файлы и директории:

`darumakal: -wxrwx-wx`

`wurmple: rwx-wx-wx`

`chinchou: права 307`

`houndour1: rw----r--`

Решение. Создадим последовательно необходимые файлы и директории в домашнем каталоге и назначим им права. Один из множества возможных вариантов.

```
cd /home/s113369; umask 022
```

```
mkdir lab0; cd lab0
```

```
mkdir -p darumakal/wurmple darumakal/chinchou
```

```
echo "Способности Howl Smog Roar Bite Odor Sleuth" > houndour1
```

```
echo "Beat Up Fire Flamethrower Crunch" >> houndour1
```

```
echo "Nasty Plot Inferno" >> houndour1
```

```
chmod go=wx darumakal/wurmple
```

```
chmod 307 darumakal/chinchou
```

```
chmod g-r houndour1
```

```
chmod 373 darumakal #должно быть последним, чтобы корректно  
#установились вложенные права
```

Задание 2. Перевести заданные числа ( $X=31681_{10}; Y=13825_{10}$ ) в 16-ричную систему счисления и представить разрядной сетке памяти БЭВМ. Вычислить значение  $R=X+Y$ , результат представить как число со знаком в 16-ричной системе счисления разрядной сетке БЭВМ. Укажите, произошло ли переполнение при операции? Был ли перенос?

Решение. Перевод чисел в шестнадцатеричную систему счисления и упаковка

исходных чисел в разрядную сетку БЭВМ (см. Приложение В, раздел 1.4) следующая:  $X=7BC1_{16}$ ,  $Y=3601_{16}$ . Результат сложения  $R=7BC1_{16}+3601_{16}=B1C2_{16}$ . В ответе необходимо привести последовательность перевода, упаковки в дополнительный код (при необходимости) и сложения в столбик. Разрешается использовать промежуточное представление в виде двоичного числа.

Перенос в 17 разряд (формирование признака C) не произошел потому, что результат меньше 65535. Переполнение (признак результата V) произошло потому, что результат больше максимально допустимого положительного числа в знаковом представлении (32767). В знаковом представлении результат ( $-20030_{10}$ ) является отрицательным числом, что является ошибкой, т. к. ведет к потере ожидаемого результата сложения (45506).

**Задание 3.** Какую операцию выполняет команда ADD AD? Перечислите все результаты этой команды.

**Решение.** Команда ADD AD относится к адресным командам с прямой абсолютной адресацией и складывает содержимое указанной в коде команды ячейки с аккумулятором (AC).

Результатом команды являются измененные значения регистров и ячеек памяти, а именно: счетчик команд будет содержать адрес следующей выполняемой команды; регистр AC — сумму содержимого AC до исполнения команды и ячейки ADA; AR и DR — содержать число AD; CR содержать код команды 40AD; в BR будет содержимое IP до выполнения команды; будут установлены признаки результата NZVC в PS по вычисленной сумме.

## **Раздел 2. Введение в базовую ЭВМ**

В рамках этого раздела студент должен изучить состав, структуру и принцип функционирования БЭВМ на уровне машинных команд, систему команд БЭВМ, детальную последовательность исполнения команд с прямой и косвенной адресациями, подпрограммы, основные подходы, применяемые для низкоуровневой обработки данных.

### **Лабораторная работа №2. Исследование работы БЭВМ**

**Цель работы** - изучение приемов работы на базовой ЭВМ и исследование порядка выполнения арифметических команд и команд пересылки.

**Задание.** По выданному преподавателем варианту определить функцию, вычисляемую программой, область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы, предложить вариант с меньшим числом команд. При выполнении работы представлять результат и все операнды арифметических операций знаковыми числами, а логических операций набором из шестнадцати логических значений.

**Подготовка к выполнению работы.** Познакомиться с устройством и системой команд базовой ЭВМ (см. приложение В, п.п. 1.1 — 1.6), порядком выполнения машинных команд (п.1.9), инструкцией по работе с моделью базовой ЭВМ (см. приложение Г). Изучить представления в БЭВМ числовых и логических значений.

**Порядок выполнения работы.** Восстановить текст заданного варианта программы, отделить ячейки данных от кода программы, написать назначение программы и реализуемую функцию, которую представить в виде формулы.

Во время допуска к работе получить у преподавателя исходные данные для переменных, согласовать вариант программы для исполнения, занести в память базовой ЭВМ заданный вариант программы и, выполняя ее по командам, заполнить таблицу трассировки выполненной программы. Занесение программы с данными, а также запуск программы в пультовом режиме продемонстрировать преподавателю.

**Содержание отчета по работе.** В дополнение к общим обязательным требованиям, отчет должен содержать:

1. Текст исходной программы по следующей форме:

Оформление текста программы для л/р №2-5.

Таблица 2.1

Адрес	Код команды	Мнемоника	Комментарии
021	4015	ADD 15	Добавить содержимое ячейки памяти 15 к аккумулятору

2. Описание программы:

- назначение программы и реализуемые ею функция (формула);
- область представления и область допустимых значений исходных данных и результата;
- расположение в памяти ЭВМ программы, исходных данных и результатов;
- адреса первой и последней выполняемой команд программы.

3. Таблица трассировки должна быть представлена в соответствии с форматом:

Форма таблицы трассировки выполнения команд.

Таблица 2.2

Выполняемая команда		Содержимое регистров процессора после выполнения команды.								Ячейка, содержимое которой изменилось после выполнения команды	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
xxx	xxxx	xxx	xxxx	xxxx	xxxx	xxx	xxxx	xxxx	xxxx	xxx	xxxx

4. Вариант программы с меньшим числом команд.

Контрольные вопросы:

1. Форматы представления, области представления и области допустимых значений в БЭВМ для знаковых и беззнаковых чисел с фиксированной точкой и логических значений.
2. Представление чисел в разрядной сетке в прямом, обратном и дополнительном кодах.
3. Адресные и безадресные команды БЭВМ.
4. Описание команды находящейся, по указанному адресу: наименование, назначение, тип команды и вид адресации. Количество и название машинных циклов, потактовое выполнение команды.
5. Какую формулу реализует программа? Как можно упростить программу?
6. Где находятся аргументы программы? Где находится результат? Как они представлены? Какие дополнительные ячейки использует программа? Для чего?
7. Какое количество обращений к ячейкам памяти при выполнении безадресной команды? На каких циклах оно выполняется?