МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И. С. ТУРГЕНЕВА»

Кафедра программной инженерии

**КОНТРОЛЬНАЯ РАБОТА**

по дисциплине: «Архитектура ЭВМ и систем»

Вариант реферативной части № 20

Вариант практической части № 8

Студент: Кожухова О.А. Шифр: 170582

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверила: Конюхова О.В.

Отметка о зачете: \_\_\_\_\_\_\_\_ Дата: «\_\_\_\_» \_\_\_\_\_\_ 20\_\_ г.

Орел, 2017 г.

**Содержание**

[Введение 3](#_Toc497507806)

[Программное обеспечение ЭВМ 4](#_Toc497507807)

[Виды программного обеспечения 5](#_Toc497507808)

[Понятие операционной системы 6](#_Toc497507809)

[Основные классификации ОС 8](#_Toc497507810)

[Классификация операционных систем по семействам 10](#_Toc497507811)

[Заключение 12](#_Toc497507812)

[Список литературы 13](#_Toc497507813)

[Задача 14](#_Toc497507814)

## Введение

В основу работы ЭВМ положен программный принцип управления, состоящий в том, что машина выполняет действия по заранее заданной программе. Этот принцип обеспечивает универсальность использования ЭВМ: в определенный момент времени решается задача соответственно выбранной программе.

Для нормального решения задач на ЭВМ нужно, чтобы программа была отлажена, не требовала доработок и имела соответствующую документацию. Поэтому относительно работы на машине часто используют термин программное обеспечение (ПО, software), под которым понимают совокупность программ, выполняемых вычислительной системой.

Целью данной работы является рассмотрение программного обеспечения ЭВМ, а также видов, характеристик и особенностей операционных систем.

## Программное обеспечение ЭВМ

Программное обеспечение – комплекс программ обеспечивающих обработку или передачу данных предназначенных для многократного использования и применения разными пользователями;

- совокупность программ системы обработки информации и программных документов, необходимых для их эксплуатации;

- совокупность программ, выполняемых вычислительной системой.

Программа – это упорядоченные последовательности команд. Конечная цель любой компьютерной программы – управление аппаратными средствами.

К программному обеспечению относится также вся область деятельности по проектированию и разработке ПО:

* технология проектирования программ;
* методы тестирования программ;
* методы доказательства правильности программ;
* анализ качества работы программ;
* документирование программ;
* разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и многое другое.

Программное обеспечение в настоящее время составляет сотни тысяч программ, которые предназначены для обработки самой разнообразной информации с самыми различными целями.

## Виды программного обеспечения

Все программы, работающие на компьютере, можно условно разделить на три вида:

1. прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;
2. системные программы, предназначены для управления работой вычислительной системы, выполняют различные вспомогательные функции, например: управление ресурсами компьютера, создание копий используемой информации, проверка работоспособности устройств компьютера, выдача справочной информации о компьютере и др.;
3. инструментальные программные системы, облегчающие процесс создания новых программ для компьютера.

## Понятие операционной системы

Операционная система - это комплекс взаимосвязанных системных программ, функциями которого является контроль использования и распределения ресурсов вычислительной системы и организация взаимодействия пользователя с компьютером.

Система загружается при включении компьютера. Она производит диалог с пользователем, осуществляет управление компьютером, его ресурсами (оперативной памятью, местом на дисках и т.д.), запускает другие (прикладные) программы на выполнение. Операционная система обеспечивает пользователю и прикладным программам удобный способ общения (интерфейс) с устройствами компьютера.

Основная причина необходимости операционной системы состоит в том, что элементарные операции для работы с устройствами компьютера и управления ресурсами компьютера - это операции очень низкого уровня, поэтому действия, которые необходимы пользователю и прикладным программам, состоят из нескольких сотен или тысяч таких элементарных операций. В составе ОС различают 3 группы компонентов:

1. ядро, содержащее планировщик; драйверы устройств, непосредственно управляющие оборудование; сетевая подсистема, файловая система;
2. системные библиотеки;
3. оболочка с утилитами.

Основные функции операционных систем:

1. Загрузка приложений в оперативную память и их выполнение;
2. Стандартизированный доступ к периферийным устройствам;
3. Управление оперативной памятью;
4. Управление доступом к данным на энергонезависимых носителях;
5. Пользовательский интерфейс;
6. Сетевые операции;
7. Параллельное или псевдопараллельное выполнение задач (многозадачность);
8. Взаимодействие между процессами: обмен данными, взаимная синхронизация;
9. Разграничение прав доступа и многопользовательский режим работы (авторизация, аутентификация).

## Основные классификации ОС

По назначению:

1. Системы общего назначения. (Подразумевает ОС, предназначенные для решения широкого круга задач, включая запуск различных приложений, разработку и отладку программ, работу с сетью и мультимедиа.);

2. Системы реального времени. (Предназначены для работы в контуре управления объектами.);

3. Прочие специализированные системы. (Это различные ОС, ориентированные на эффективное решение определенного класса, с большим или меньшим ущербом для прочих задач.).

По характеру взаимодействия с пользователем:

1. Пакетные ОС, обрабатывающие заранее подготовленные задания;

2. Диалоговые ОС, выполняющие задания пользователя в интерактивном режиме;

3. ОС с графическим интерфейсом;

4. Встроенные ОС, не взаимодействующие с пользователем.

По числу одновременного выполнения задач:

1. Однозадачные ОС. (В таких систем ах в каждый момент времени может существовать не более чем один пользовательский процесс. Однако, одновременно с этим, могут работать системные процессы);

2. Многозадачные ОС. (Они обеспечивают параллельное выполнение некоторых пользовательских процессов. Реализация многозадачности требует значительного усложнения алгоритмов и структур данных, используемых в системе.).

По числу одновременных пользователей:

1. Однопользовательские ОС. (Для них характерен полный пользовательский доступ к ресурсам. Подобные системы приемлемы в основном на изолированных компьютерах.);

2. Многопользовательские ОС. (Их важной компонентой являются средства защиты данных и процессов каждого пользователя, основанные на понятии владельца ресурса и на точном указании прав доступа, предоставленных каждому пользователю системы.).

По аппаратурной основе:

1. Однопроцессорные ОС;

2. Многопроцессорные ОС. (В задачи такой системы входит эффективное распределение выполняемых заданий по процессорам и организация согласованной работы всех процессоров.);

3. Сетевые ОС. (Они включают возможность доступа к другим компьютерам локальной сети, работы с файловыми и другими серверами.);

4. Распределенные ОС. (Распределенная система, используя ресурсы локальной сети, представляет их пользователю как единую систему, не разделенную на отдельные машины.).

По способу построения:

1. Микроядерные;

2. Монолитные.

## Классификация операционных систем по семействам

Операционные системы семейства OS/2.

OS/2 – семейство многозадачных операционных систем с графическим интерфейсом, есть версии для многопроцессорных машин. OS/2 создавалась для собственных нужд и задач фирмы IMB. OS/2 использовалась IMB в качестве основы некоторого числа программных решений, таких как комментаторские системы олимпийских игр, программное обеспечение для банков. Под нее практически не существует программного обеспечения.

Операционные системы семейства UNIX.

Первая система UNIX была разработана в 1969 г. в подразделении Bell Labs компании AT&T. С тех пор было создано большое количество различных UNIX-систем. Все ОС, относящиеся к этому семейству, являются многозадачными, многопользовательскими, с графическим интерфейсом, обеспечивают достаточную надежность и защиту данных. Эти ОС ставятся на различные аппаратные платформы.

Отличительные признаки UNIX-систем включают в себя: использование простых текстовых файлов для настройки и управление системой, широкое применение утилит, запускаемых в командной строке, взаимодействие с пользователем посредством виртуального устройства – терминалом, использование конвейеров из нескольких программ, каждая из которых выполняет одну задачу, предоставление физических и виртуальных устройств и некоторых средств межпроцессорного взаимодействия как файлов.

Операционные системы семейства Linux.

Linux является одной из распространенных систем версий UNIX. Она может организовать работу как рабочих станций, так и сервера. Поддерживает технологию Plug & Play (стандарт аппаратной и программной архитектуры, который делает возможным распознавание устройств).

Это многозадачная и многопользовательская ОС для бизнеса, образования и индивидуального программирования, ориентирована на работу в сети.

Одним из достоинств Linux можно считать высокую скорость работы. Эта ОС может работать на машинах не очень большой мощности. Второе достоинство заключается в том, что она может применяться как для различных типов серверов, так и для настольных компьютеров.

Операционные системы семейства Windows.

Все ОС семейства Windows являются многозадачными системами с графическим интерфейсом. Они работают на платформах x86, x86-64, IA-64, ARM..

Одним из достоинств ОС семейства Windows является поддержка технологии Plug & Play. Эта технология упрощает для пользователя подключение разных внешних устройств. Еще одним достоинством этих ОС является их переносимость: за счет специальных модулей осуществляется связь ОС с разным аппаратным обеспечением.

ОС семейства Windows реализует метод многозадачности с вытеснением. Это позволяет снять приложение в случае его зависания. Также эти ОС поддерживают технологию OLE (ObjectLinkingEmbedding). OLE – стандарт, позволяющий создавать различные составные документы.

В интерфейсе ОС семейства Windows реализована объектная модель. Также они поддерживают работу ПК в сети.

## Заключение

В ходе работы было дано определение программному обеспечению, приведены его виды, а также дано определение операционной системе, выделены ее основные компоненты и отмечены функции, приведены основные классификации ОС, подробно рассмотрена классификация ОС по семействам.

## Список литературы

1. Острейковский В.А. Информатика: учебник для вузов / В.А. Острейковский, М.: Высшая школа, 2011. - 511 с.
2. Степанов А.Н. Информатика: учебное пособие для вузов / А.Н. Степанов, 4-е изд., СПб.: Питер, 2012. - 684 с.
3. Таненбаум Э. Современные операционные системы / Эндрю Таненбаум, 2-е изд., СПб.: Питер, 2009, 2010. - 1038 с.

## Задача

Написать программу на языке Ассемблера, которая позволяет ввести с клавиатуры массив целых чисел из 7 элементов, найти и вывести на экран первый положительный элемент и его порядковый номер; в противном случае, вывести на экран соответствующее сообщение. Ноль считается положительным элементом.

; Индексы массива считаются с 0003h

; Допустимые к вводу целые числа -9..9

S\_S segment stack

db 100 dup (?) ; Резервируем в стеке 100 байт

S\_S ends

D\_S segment

a db 7 dup (1) ; Вводимый массив целых чисел

msg\_not\_found db 0dh, 0ah, 'Number not found !$' ; Сообщение, ;выводимое если не найдено число >= 0

msg\_found db 0dh, 0ah, 'Array[i] = $' ; Сообщение, выводимое если ;число найдено

D\_S ends

C\_S segment

assume ss:S\_S, cs:C\_S, ds:D\_S, es:D\_S

begin:

mov ax, D\_S

mov ds, ax ; Подготовка регистра сегмента данных

mov ax, 0003h

int 10h ; Очистка экрана

mov cx, 7 ; Повторяем цикл loop 7 раз, т.к. в массиве 7 чисел

mov si, 0 ; si - индекс элемента массива, начиная с нуля

INPUT\_A: ; Считывание символа с клавиатуры, al - код введёного символа

mov ah, 01h

int 21h

cmp al, ' ' ; Если ввели пробел, то ввод следующего символа

je INPUT\_A

cmp al, '-' ; Если ввели минус '-', число отрицательно

jne CMPN1 ; Если был введён не минус, переходим к следующим ;проверкам

mov a[si], -1 ; Иначе, делаем элемент массива отрицательным

jmp INPUT\_A ; Переходим к вводу следующего символа

CMPN1: ; Проверяем, является ли символ числом, если нет, то ;выходим из программы

cmp al, '0'

jb EXIT

cmp al, '9'

jg EXIT

; В любом другом случае было введено число

Sub al, '0' ; Преобразуем символ в число: al - код '0' = число

; Запись числа в массив

mov bl, al

mov al, a[si]

imul bl

mov a[si], al ; a[si] = a[si] \* bl (a[si] может быть = -1, если был ;введён '-', изначально = 1)

inc si ; Переходим к следующему числу

loop INPUT\_A

; Поиск первого положительного элемента массива

mov si, 0 ; si - индекс элемента массива, начиная с нуля

mov cx, 7 ; Повторяем цикл loop 7 раз, т.к. в массиве 7 чисел

FIND\_POSITIV:

; Проверка очередного числа массива на >= 0

cmp a[si], 0

jge FIND\_POSITIV\_OK ; Выходим из цикла, если нашли ;положительное число

inc si ; Переходим к следующему элементу

loop FIND\_POSITIV

; Сюда попадём, если не нашли подходящее число

; Выведем сообщение о том, что число не найдено

mov ah, 09h ; Функция вывода строки

mov dx, offset msg\_not\_found ; Смещение строки-сообщения

int 21h

jmp EXIT

FIND\_POSITIV\_OK:

; Сюда переходим, если нашли положительное число

; si - индекс найденного числа, a[si] - число

mov cl, a[si] ; Сохраняем найденное число

add cl, '0' ; Преобразуем число в символ: сl + код '0' = число в ;виде символа

add si, '0' ; Преобразуем индекс в символ: si + код '0' = индекс в ;виде символа

; Записываем индекс найденого элемента в строку msg\_found вместо буквы i 'Array[вместо i - индекс] = '

mov ax, D\_S

mov es, ax

mov ax, si

mov di, offset msg\_found + 8

stosb ; Команда записывает байт из al по адресу es:[di]

; Выводим сообщение об успехе и индекс элемента

mov ah, 09h ; Функция вывода строки

mov dx, offset msg\_found ; Смещение строки-сообщения

int 21h

; Вывод числа

mov ah, 02h ; Функция вывода символа

mov dl, cl ; dl - символ для вывода, т.е. найденное число

int 21h

EXIT: ; Завершение программы

mov ax, 4c00h

int 21h

C\_S ends

end begin

**Алгоритм:**

1. Ввод массива целых чисел заданной длины. Допустимый диапазон значений: [-9;9].
   1. Считываем очередной символ с клавиатуры.
   2. Если считанный символ = ‘ ‘ (пробел), то переходим к пункту 1.a.
   3. Если считанный символ = ‘-‘ (минус), то очередной элемент массива будет отрицательным.
   4. Если считанный символ не равен ‘ ‘ или ‘-‘ или ‘0’..’9’, то ошибка ввода и завершение программы.
   5. В любом другом случае считанный символ - цифра '0'..'9'.

Преобразуем введённый символ в число: код\_символа - ‘0’ = число.

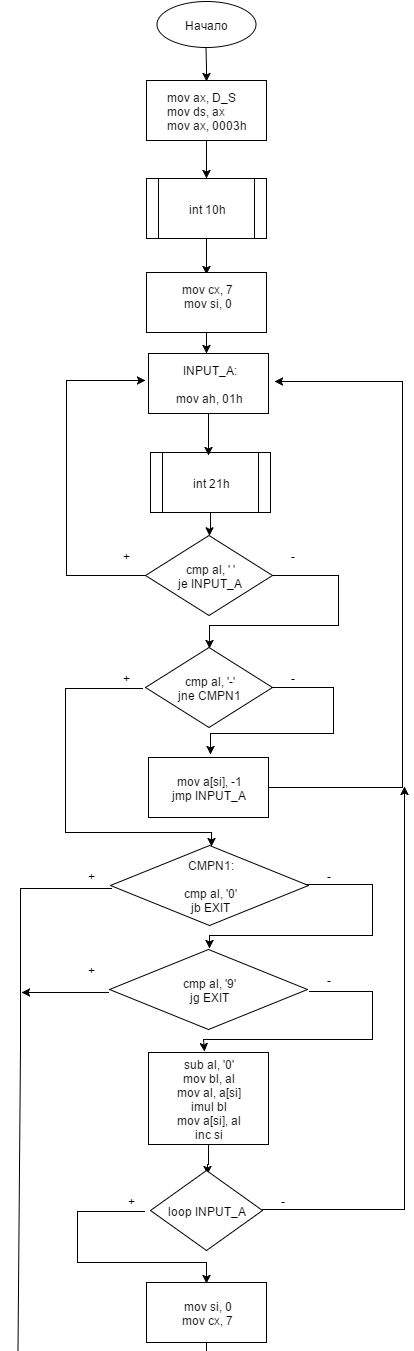
Заносим полученное число в очередной элемент массива.

* 1. Переходим к вводу следующего символа 1.a , пока не будут введены все числа.

1. Поиск первого положительного элемента массива и его индекса.

Пока не просмотрим весь массив или не найдём положительный элемент массива, в цикле проверяем каждый элемент с начала массива на >= 0.

1. Если не был найден положительный элемент массива, выводим соответствующее сообщение пользователю.
2. Если был найден положительный элемент массива, выводим его значение и индекс.
   1. Преобразуем число и его индекс в массиве следующим способом:  
      число + ’0’ = ‘код символа, соответствующий числу’.
   2. Выводим на экран число и его индекс в массиве.

****

