

ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

Файл – совокупность *однотипных данных*, которой присвоено имя и которая находится в памяти компьютера (оперативной или внешней)

Два вида файлов:

- а) ***внешний файл*** - совокупность данных на дисках (или др. носителях : MT, DF, Flash) , а так же поток данных, выводимый на печатающее ус-во или экран и вводимых с клавиатуры.
- б) ***внутренний файл*** - это подстрока, символьная переменная, эл-т символьного массива либо символьный или не символьный массив. (для ФОРТРАН 90 и ↑ есть особенности: символьным производным типом)

Примеры режимов доступа к внешним файлам для ФОРТРАН 2003 и ↑ :

- 1) Асинхронный ввод-вывод - спецификатор ASYNCHRONOUS = ' YES '
- 2) Поточковый ввод-вывод - спецификатор ACCESS = ' STREAM '

Понятие **файловый указатель** : текущее положение в файле обозначается файловым указателем.

Операция в файле выполняется там, где расположен указатель. Указатель перемещается по файлу при выполнении операций ввода-вывода, а также с помощью специальных команд.

Файл состоит из ***записей***, расположенных последовательно. Все записи в файле имеют один тип. Именно записи определяют организацию файла , его внутреннее устройство.

Три основных типа записей : 1.) Форматные записи ; 2.) Неформатные записи ; 3.) Запись “конец файла “

По **методу доступа** различаются два типа файлов :

- а) ***файлы последовательного доступа*** записи расположены последовательно. Доступ к данным по порядку, запись за записью . Новая запись может быть добавлена только в конец файла.
- б) ***файлы прямого доступа*** представляют собой единый участок памяти, в котором хранится часть файла. Участки памяти имеют фиксированную длину и пронумерованы от 1 до N. Данные можно считывать и записывать в любом порядке. Используются, когда необходим доступ к данным в произвольном порядке.

ОПЕРАТОРЫ ВВОДА-ВЫВОДА

BACKSPACE — устанавливает файл в начало предыдущей записи;
CLOSE — отсоединяет устройство;
ENDFILE — выводит запись "Конец файла";
INQUIRE — выдает информацию о свойствах устройства или
именованного файла;
LOCKING — управляет доступом к файлу прямого доступа;
OPEN — присоединяет устройство к файлу (открывает файл);
PRINT — выводит данные на стандартное устройство;
READ — вводит данные;
REWIND — устанавливает файл в его начальную точку;
WRITE — выводит данные.

В операторах ввода-вывода свойства файла и способ передачи данных задаются параметрами. Во многих операторах используются следующие параметры:

FILE — задает имя файла;
UNIT — задает идентификатор устройства;
ACCESS — определяет метод доступа: последовательный или
прямой;
FORM — задает вид записей файла: форматные, бесформатные
или двоичные;
список-в/в — задает объекты программы для передачи;
IOSTAT, ERR и END — позволяют управлять обработкой ошибок
ввода-вывода и записи "конец файла";
MODE и SHARE — задают операции ввода-вывода, допустимые
при совместной обработке файла;
FMT — указывает способ передачи форматных записей.

ИМЯ ФАЙЛА

Имя внутреннего файла – имя подстроки, символьной переменной, эл-та символьного массива, символьного или несимвольного массива.

Имя внешнего файла - любое имя для именования файлов в операционной системе. Есть специальные : CON , PRN , NUL.

Способы задания имени внешнего файла: 1) в параметре FILE оператора OPEN ; 2) по умолчанию (если в OPEN отсутствует параметр FILE) – Временный файл (STATUS='SCRATCH') с именем ZZхууууу, где х – 0, а, в,

Пример. Второй временный файл может иметь имя ZZa12345 (число 12345 назначается ОС)

ИДЕНТИФИКАТОР УСТРОЙСТВА

Обращение к файлу в операторах I / O происходит по идентификатору ус-ва , задается в UNIT (значение -32767 до 32767 или *)

Три способа присоединения внешнего файла к устройству

1) Явное присоединение

Пример. OPEN (UNIT = 9, FILE = ' FILE9 '
READ (9, *) а, b, с

CLOSE – отменяет присоединение или завершение пр-мы.

2) Неявное присоединение

Происходит по первому оператору READ или WRITE, если не используется OPEN. Отсоединение файла автоматически по завершении программы .

3) Предварительное присоединение

Внешний файл присоединяется по умолчанию к устройству в начале выполнения программы.
Присоединяются только четыре устройства : →

Устройство	Внешний файл
* (звездочка)	Клавиатура, экран
0	Клавиатура, экран
5	Клавиатура
6	Экран

Пример внутреннего файла

```
CHARACTER*9 file4  
num=4  
WRITE(file4,30) num  
30 FORMAT('FT',I3.3,'.DAT')  
WRITE(*,*) file4  
END
```

В первом операторе WRITE символьная переменная *file4* является внутренним файлом, в который выводится значение *num* под управлением оператора FORMAT. По второму WRITE на экран выводится значение этой переменной : FT004.DAT.

МЕТОД ДОСТУПА К ФАЙЛУ И ВИД ЗАПИСЕЙ

Файл состоит из записей. Последовательный доступ – ACCESS = ' SEQUENTIAL ' ; Прямой доступ - ACCESS = ' DIRECT ' .
Метод доступа задается в операторе OPEN. По умолчанию действует - ' SEQUENTIAL ' .

- 1) *Последовательный доступ* (Доступ к внутреннему файлу и на устройствах CON, PRN – только последовател.)
- 2) *Прямой доступ к файлу* - позволяет считывать и записывать в произвольном порядке.

К файлу, присоединенному для прямого доступа, можно обращаться операторами I/O последовательного доступа.

Пример 1.

```
...  
OPEN(8,FILE='DAT',ACCESS='DIRECT',RECL=20,FORM='FORMATTED')  
READ(8,'(4I5)',REC=5) i,j,k,l  
READ(8,'(4I5)') l,m,n,nn  
...
```

Из файла DAT по оператору READ прямого доступа считывается 5-ая запись , а по оператору READ последовательного доступа – 6-ая запись.

В зависимости от представления данных в записях различаются записи :

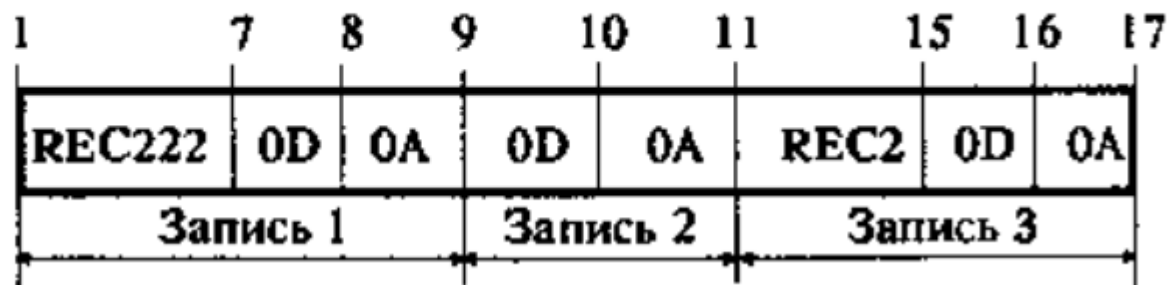
- форматные FORM = ' FORMATTED '
- бесформатные FORM = ' UNFORMATTED '
- двоичные FORM = ' BINARY '

Каждая запись внешнего форматного файла (прямого и последовательного) заканчивается признаком конца данных: символами возврата каретки (0D) и перевода строки (0A)

Пример 2.

```
OPEN (7,FILE='FSQ')
WRITE (7,'(A,I3)') 'REC', 222
WRITE (7,'( )')
WRITE (7, '(''REC2 ''')')
CLOSE (7)
```

В результате выполнения в файл выводятся три записи:



3) **Бесформатный последовательный файл** состоит из записей переменной длины, определяемой списком ввода-вывода. 1-ый байт всегда содержит десятичное 75 , последний байт – десятичное 130

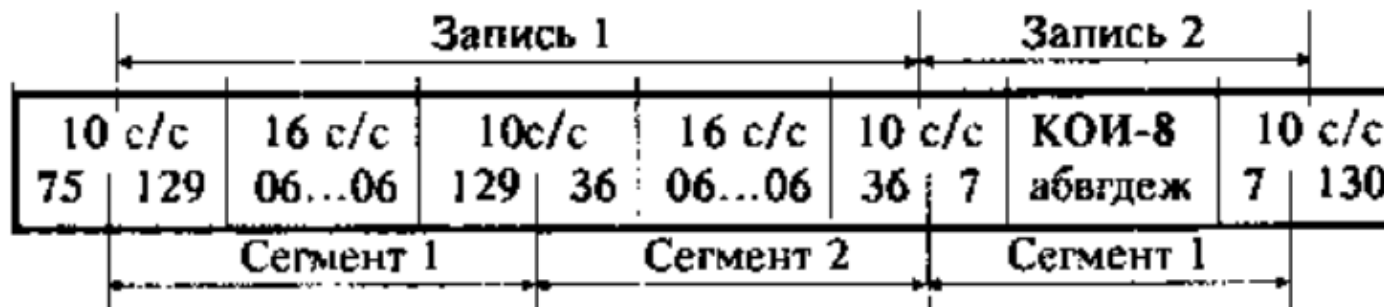
Пример 3.

```

CHARACTER symbol(7)
INTEGER*1 array(164)
DATA array/164*6/, symbol/'а','б','в','г','д','е','ж'/
OPEN (8,FILE='UFSQ',FORM='UNFORMATTED')
WRITE (8) array
WRITE (8) symbol
CLOSE (8)
END

```

В результате выполнения программы в файл выводятся две записи:



4) Бесформатный прямой файл - все записи имеют одинаковую длину, заданную в параметре RECL оператора OPEN .Между записями нет разделителей.

Пример 4.

```
OPEN (8, FILE='UNFDIR',FORM='UNFORMATTED',
*ACCESS='DIRECT',RECL=10)
WRITE (8,REC=3) 255,'абвгде'
WRITE (8,REC=1) 255
CLOSE (8)
END
```

В файл выводятся первая и третья записи. Значение второй записи в этой программе не устанавливается и будет неопределенным, если файл новый. В результате выполнения программы файл имеет вид:



5) Двоичный последовательный файл - последовательность двоичных записей (на сегменты не разбивается).

Пример 5.

```

INTEGER*1 bel(4)
INTEGER arr(3)
CHARACTER cvar*4
DATA bel /4*5/
DATA cvar/'6666'/.arr/3*'aaaa'/
OPEN (8, FILE='BSQ',FORM='BINARY')
WRITE (8) arr,cvar
WRITE (8) 'vvvvv','ttttttt'
WRITE (8) bel
CLOSE (8)
END

```

В результате выполнения программы в файл выводятся следующие данные:

1	13	17	22	30	34
КОИ-8 аааааааааа 6666		КОИ-8 vvvvv ttttttt		16 с/с 05050505	
Запись 1		Запись 2		Запись 3	

6) Двоичный прямой файл – похож на бесформатный прямой файл, но есть *отличия*:

- а) если длина выводимых данных меньше длины записи, запись не дополняется нулями. Неиспользованная часть записи содержит неопределенные значения;
- б) если данных передается больше, чем содержит одна запись, передача будет продолжена со следующей записи.

Пример 6.

```
OPEN (8,FILE='BDIR',RECL=10,FORM='BINARY',ACCESS='DIRECT')
WRITE (8,REC=1) 'aaaaaaaaab6666'
WRITE (8,REC=3) 1,2
WRITE (8,REC=4) 'вв'
CLOSE (8)
END
```

В результате выполнения программы в файл выводятся следующие четыре записи:

1	11	21	31	41
КОИ-8 аааааааааа	КОИ-8 66666.....	16 с/с 01000000002000000..	КОИ-8 вв.....	
Запись 1	Запись 2	Запись 3	Запись 4	

СПИСОК ВВОДА ВЫВОДА

Список ввода-вывода задает объекты программы, значения которых участвуют в передаче данных.

Элементы списка :

- 1) переменная, эл-т массива, подстрока;
- 2) массив (все эл-ты, расположенные в памяти);
- 3) выражение (любого типа можно использовать только в операторах WRITE , PRINT;
- 4) список с циклом вида – (список I/O, переменная-цикла=начало, конец [, приращение])

Пример

```
.....  
      WRITE(*,99) cnt,(arg3(k),k=1,cnt)  
99  FORMAT(1X,2I5)
```

ОБРАБОТКА ОШИБКИ И ЗАПИСИ “ КОНЕЦ ФАЙЛА ”

Если при выполнении оператора I/O возникает ошибка или при вводе считывается запись “ конец файла “ , то выдается сообщение об ошибке и программа **прекращает** работу.

Используя в операторах I/O параметры : IOSTAT = *целая-переменная*

ERR = *err- метка*

END = *err- метка* , можно выполнить обработку таких

ситуаций и продолжить выполнение программы.

В операторе PRINT не может быть задан ни один из указанных параметров. Любая ошибка при выполнении этого оператора приводит к прекращению выполнения программы.

Оператор READ хорошо использовать с параметрами ERR , IOSTAT для обработки ошибок. Пример. 10

Пример

```
INTEGER w,v
DATA v/0/
WRITE(*,*) ' Введите w'
50 READ(*,*,ERR=100,IOSTAT=v) w
WRITE(*,*) ' Значение w:',w,' iostat = ',v
STOP
100 WRITE(*,*) ' iostat = ',v,' Введите правильное значение w:'
GOTO 50
END
```

В операторах ввода-вывода используются параметры **ERR** и **IOSTAT** для обработки неправильных значений.

СОВМЕСТНАЯ ОБРАБОТКА ФАЙЛА

Это когда одновременно к одному и тому же файлу могут обращаться из нескольких программ, выполняющихся параллельно. Доступ к такому файлу определяется в **OPEN** с помощью параметров **MODE** и **SHARE**.

MODE = ' READ ' (варианты: ' WRITE ', ' READWRITE ' – по умолчанию)

SHARE = ' COMPAT ' (варианты: ' DENYRW ', ' DENYWR ', ' DENYRD ', ' DENYNONE ')

Значения параметров **MODE** и **SHARE**, действующие в текущий момент, можно получить по оператору **INQUIRE**. В таблице приводятся значения параметров **MODE** и **SHARE**, которые можно указывать при открытии файла для совместной обработки.

Таблица

Значение параметров MODE и SHARE

В первой программе файл открыт с параметрами		В другой программе для совместной обработки файл можно открыть с параметрами	
MODE	SHARE	MODE	SHARE
READWRITE READ WRITE	COMPAT	READWRITE READ WRITE	COMPAT
READWRITE READ WRITE	DENYRW	Параллельная обработка не допускается	
READWRITE	DENYWR	READ	DENYNONE
READ	DENYWR	READ	DENYNONE DENYWR
WRITE	DENYWR	READ	DENYNONE DENYRD
READWRITE	DENYRD	WRITE	DENYNONE
READ	DENYRD	WRITE	DENYNONE DENYWR
WRITE	DENYRD	WRITE	DENYNONE DENYRD
READWRITE	DENYNONE	READWRITE READ WRITE	DENYNONE
READ	DENYNONE	READWRITE READ WRITE	DENYNONE DENYWR
WRITE	DENYNONE	READWRITE READ WRITE	DENYNONE DENYRD

Оператор INQUIRE

В операторе очень много параметров (22 параметра , например: ACCESS, BINARY, FORMATTED, EXIST и др.). Предназначен для получения информации о характеристиках и текущем состоянии указанного **внешнего** файла или устройства. В операторе может быть задан только один из параметров: либо → UNIT , либо → FILE.

Пример.

```
CHARACTER*8 fname/'text.dat'/  
LOGICAL ext  
INQUIRE(FILE=fname,EXIST=ext)
```

C Проверка существования файла

```
IF (.not.ext) THEN  
WRITE(*,'(2A/)') ' *** Невозможно найти файл ',fname  
ENDIF
```

Оператор INQUIRE используется для определения, существует ли файл с именем text.dat. Если файла нет, выдается сообщение.

ВНУТРЕННИЙ ФАЙЛ

Внутренний файл состоит из форматных записей и предназначен для передачи данных из одной области ОП в другую. Передача данных осуществляется только операторами форматного I/O *последовательного доступа* READ , WRITE (I/O , управляемый списком, **не допускается**)

Создание спецификации файла

Пример.

```
CHARACTER fname*64
fname='FILE .DAT'
WRITE(*,*) ' Введите номер файла (1 — 3 цифры) '
READ(*,*) id
WRITE(fname(5:7),100) id
100 FORMAT(I3.3)
OPEN(1,FILE=fname,STATUS='NEW')
...
END
```