

```
gest(str);
```

Данная функция считывает символы до тех пор, пока пользователь не нажмет клавишу Enter, то есть введет символ перевода строки ‘\n’. Затем она записывает вместо символа ‘\n’ символ ‘\0’ и передает строку вызывающей программе.

Для вывода строк на экран помимо функции printf() можно использовать также функцию puts() библиотеки stdio.h, которая более проста в использовании.

Еще одной удобной функцией работы со строками является функция sprintf() библиотеки stdio.h. Ее действие аналогично рассмотренной ранее функции printf() с той лишь разницей, что результат вывода заносится в строковую переменную, а не на экран.

УДК 322

Разуев Д.А.

СТРУКТУРЫ В C/C++

БНТУ, Минск

Научный руководитель: Дробыш А.А.

Структура – это совокупность переменных, объединенных одним именем, предоставляющая общепринятый способ совместного хранения информации. Объявление структуры приводит к образованию шаблона, используемого для создания объектов структуры. Переменные, образующие структуру, называются членами структуры. (Члены структуры также часто называются элементами или полями.)

Обычно все члены структуры связаны друг с другом. Например, информация об имени и адресе, находящаяся в списке рассылки, обычно представляется в виде структуры. Следующий фрагмент кода объявляет шаблон структуры, определяющий имя и адрес.

Ключевое слово `struct` сообщает компилятору об объявлении структуры.

```
struct addr {  
    char name[30];  
    char street [40]; char city[20];  
    char state[3];  
    unsigned long int zip;  
};
```

Объявление завершается точкой с запятой, поскольку объявление структуры – это оператор. Имя структуры `addr` идентифицирует структуру данных и является спецификатором типа. Имя структуры часто используют как ярлык.

На данный момент на самом деле не создано никакой переменной. Определена только форма данных. Для объявления настоящей переменной, соответствующей данной структуре, следует написать:

```
struct addr addr_info;
```

В данной строке происходит объявление переменной `addr_info` типа `addr`. При объявлении структуры определяется переменная смешанного типа. До тех пор, пока не будет объявлена переменная данного типа, она не будет существовать.

Когда объявлена структурная переменная, компилятор автоматически выделяет необходимый участок памяти для размещения всех ее членов.

При объявлении структуры можно одновременно объявить одну или несколько переменных.

Например:

```
struct addr {  
    char name[30];  
    char street[40];  
    char city[20];  
    char state[3];  
    unsigned long int zip;} addr_info; binfo, cinfo;
```

объявляет структуру `addr` и объявляет переменные `addr_info`, `binfo`, `cinfo` данного типа.

Важно понять, что каждая вновь создаваемая структурная переменная содержит свои собственные копии переменных, образующих структуру. Например, поле `zip` переменной `binfo` отделено от поля `zip` переменной `cinfo`. Фактически, единственная связь между `binfo` и `cinfo` заключается в том, что они обе являются экземплярами одного типа структуры. Больше между ними нет связи.

Если необходима только одна структурная переменная, то нет необходимости в ярлыке структуры. Это означает, что

```
struct {  
    char name[30];  
    char street[40];  
    char city[20];  
    char state[3];  
    unsigned long int zip;} addr_info;
```

объявляет одну переменную `addr_info` с типом, определенным предшествующей ей структурой. Стандартный вид объявления структуры следующий:

```
struct ярлык {  
    тип имя переменной;  
    тип имя переменной;  
    тип имя переменной;  
} структурные переменные;
```

Ярлык – это имя типа структуры, а не имя переменной.

Структурные переменные – это разделенный запятыми список имен переменных. Следует помнить, что или ярлык, или структурные переменные могут отсутствовать, но не оба.