

# Лекция 11

Алгоритмические языки и  
программирование

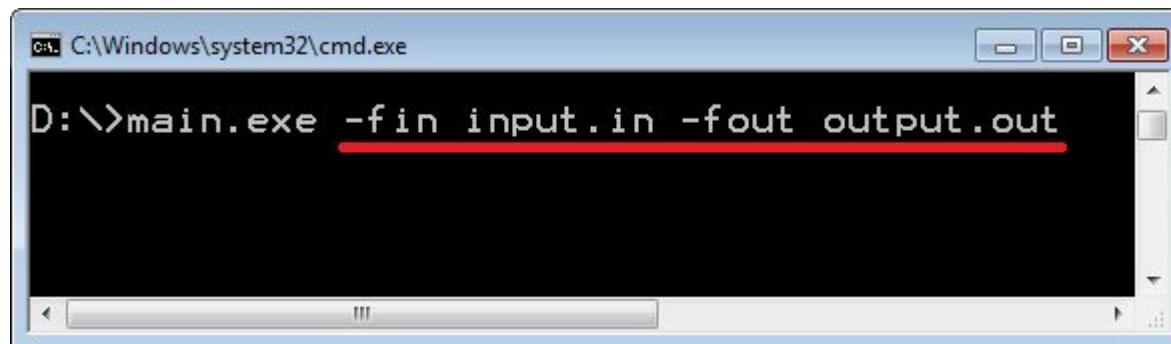
# Аргументы командной строки

# Часть 1

# Аргументы командной строки

При запуске программы через консоль возможно передать в программу данные, называемые **Аргументы командной строки**, в виде строк.

Они могут быть использованы во время работы программы



```
C:\Windows\system32\cmd.exe
D:\>main.exe -fin input.in -fout output.out
```

The image shows a screenshot of a Windows command prompt window. The title bar reads "C:\Windows\system32\cmd.exe". The command prompt shows the current directory as "D:\>" and the command "main.exe -fin input.in -fout output.out" has been entered. A red underline is drawn under the arguments "-fin input.in -fout output.out".

# Аргументы командной строки

Обратиться к аргументам командной строки в программе возможно через специальные переменные `int argc` и `char *argv[]`

`argc` – число переданных аргументов,  
`argv` – массив строк равный числу аргументов.

При вызове программы всегда есть один аргумент имя запущенной программы.

# Аргументы командной строки

Программы могут принимать аргументы. Ниже приведен пример программы `argv`, которая печатает список аргументов, которые были переданы ей в командной строке.

Пример ее работы:

```
bash$ ./argv alpha beta gamma last
```

```
Argument 0: ./argv
```

```
Argument 1: alpha
```

```
Argument 2: beta
```

```
Argument 3: gamma
```

```
Argument 4: last
```

# Аргументы командной строки

```
#include <stdio.h>
```

```
int main (int argc, char * argv[]) {
```

```
    int i;
```

```
    for( i = 0 ; i < argc; i++) {
```

```
        printf("Argument %d: %s\n", i, argv[i]);
```

```
    }
```

```
    if(argc == 1) {
```

```
        printf("Command line has no additional arguments\n");
```

```
    }
```

```
    return 0;
```

```
}
```

# Аргументы командной строки

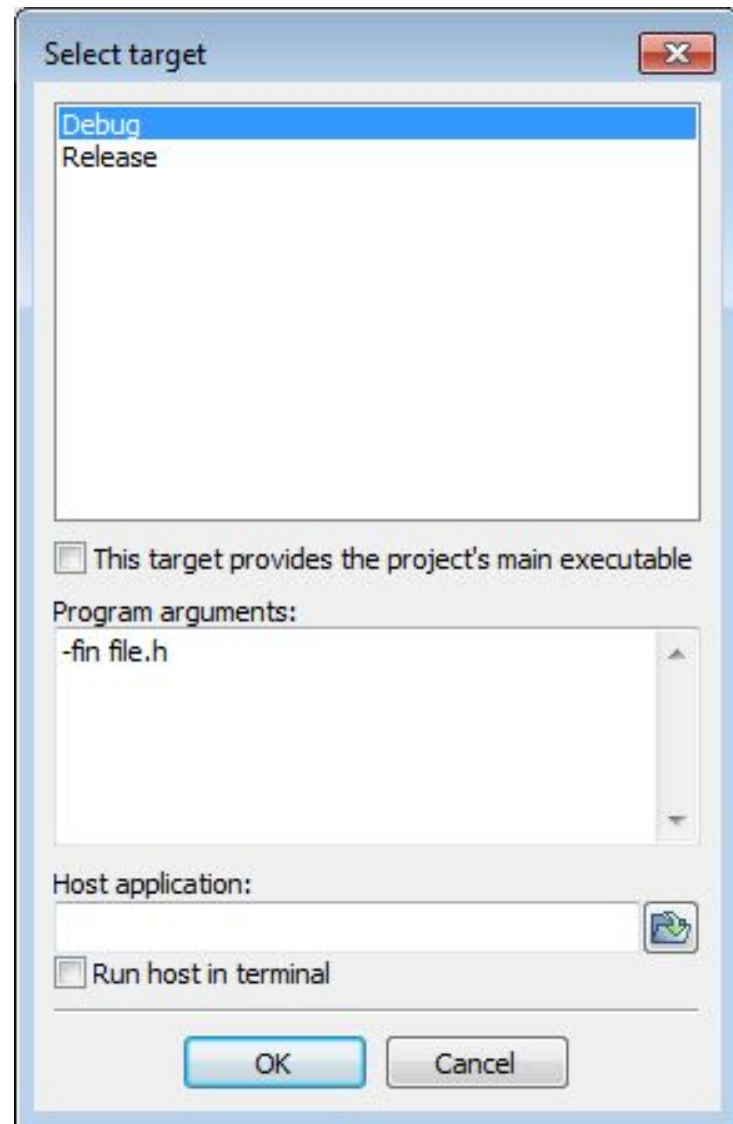
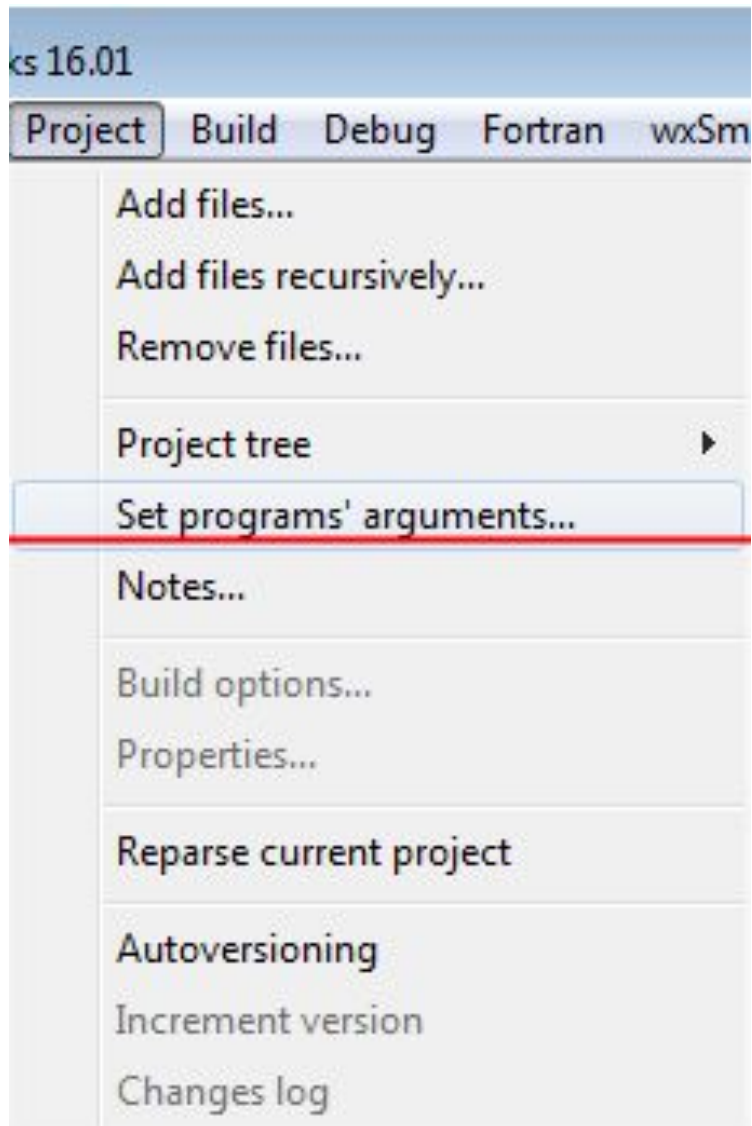
- Программа может получать данные из нескольких источников. Наиболее важные источники это:
- аргументы командной строки;
- стандартный поток ввода;
- файлы;
- средства взаимодействия процессов (signals, sockets, messages, ...).
- Научимся использовать первый из них.



# Аргументы командной строки

- Основной функцией в программе на языке Си является функция `main()`. Мы уже познакомились с простейшим способом объявления этой функции.
- Для того, чтобы ваша программа могла анализировать параметры, передаваемые ей через командную строку, функцию `main()` следует описать несколько сложнее:
- `int main(int argc, char **argv);`
- При такой записи, при вызове вашей программы, параметр `argc` будет содержать количество параметров командной строки **плюс один**. Действительные параметры нумеруются с единицы по `argc-1` включительно (т.е. всего `argc-1` параметров), параметр под номером ноль является служебным — он содержит полный путь и имя исполняемого файла вашей программы.
- Для доступа к параметру под номером `i` следует писать `argv[i]`. Выражение `argv[i]` — это строка, передаваемая в соответствии с соглашениями языка Си о строках. С ней можно проделывать все операции, как с обычной строкой.

# Аргументы командной строки



# Лабораторные работы

# Аргументы командной строки

- При вводе ключей командной строки (одного или их комбинации) —h / -? / —help. Вывести текстовое сообщение с краткой справкой о работе программы

# Аргументы командной строки

- Напишите программу, которая принимает аргумент `--q.txt` (имя файла), и выводит содержимое этого файла.

# Аргументы командной строки

- Напишите программу, которая интерпретирует свои аргументы как целые числа и выводит на стандартный поток вывода их сумму.