



ФГОБУ ВПО "СибГУТИ"
Кафедра вычислительных систем

Дисциплины
"ЯЗЫКИ ПРОГРАММИРОВАНИЯ"
"ПРОГРАММИРОВАНИЕ"

Практическое занятие №7

Вывод информации

Преподаватель:

Доцент Кафедры ВС, к.т.н.

Поляков Артем Юрьевич



Ввод данных

scanf (SCAN Formatted) – функция, обеспечивающая ввод информации с клавиатуры компьютера (стандартный ввод).

Функции **scanf** необходимо передать следующие **аргументы**:

1. Строку, содержащую шаблон текста, выводимого на экран. Шаблон может содержать:

символы разделители: пробел (' '), табуляция ('\t'), новая строка ('\n'), такой элемент соответствует любому количеству любых разделителей во входном тексте.

обычные символы (не разделители и не %): выполняется точное сопоставление со входным текстом

спецификаторы: такие же как для функции **printf**.

2. Список указателей на переменные в которые осуществляется ввод (разделитель – запятая). Для получения указателя на переменную используется операция '&', например &x – указатель на переменную x.



Справочное руководство (scanf)

```
$ man 3 scanf
```

Для работы с описанными функциями нужно подключить файл **stdio.h**

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

```
int scanf(const char *format, ...);  
int fscanf(FILE *stream, const char *format, ...);  
int sscanf(const char *str, const char *format, ...);
```

Функции ввода в стандартной библиотеке языка Си имеют схожие прототипы (формат вызова) и соответствуют функциям ввода:

scanf– ввод с клавиатуры (стандартный ввод).

fscanf– ввод из файла

sscanf– ввод из строки

В ОС Unix/Linux клавиатура – это файл, поэтому
scanf(...)= fscanf(stdin, ...)



C07.1 Ввод значения целой знаковой переменной

c06_6.c

```
#include <stdio.h>

int main()
{
    int i;
    printf("Input i value: ");
    scanf("%d", &i);
    printf("Your input was %d\n", i);
    printf("i + 5 = %d\n", i+5);
    return 0;
}
```

```
$ ./c06_6
Input i value: 10
Your input was 10
i + 5 = 15
```



C07.2 Одновременный ввод двух целых переменных (разделители)

c06_7.c

```
#include <stdio.h>
int main()
{
    int i, j;
    printf("Input i & j value: ");
    scanf("%d %d", &i, &j);
    printf("Your input was i=%d, j=%d\nInput i & j again: ", i, j);
    scanf("%d %d", &i, &j);
    printf("Your input was i=%d, j=%d\n", i, j);
    printf("i + j = %d\n", i + j);
    return 0;
}
```

```
$ ./c06_7
Input i & j value: 4 5
Your input was i=4, j=5
Input i & j again: 2
3
Your input was i=2, j=3
i + j = 5
```

Функция scanf не делает различий между разделителями и не обращает внимания на их количество.



С07.3 Использование СИМВОЛОВ-НЕ-СПЕЦИФИКАТОРОВ в форматной строке

c06_8.c

```
#include <stdio.h>
int main()
{
    int i = 0, j = 0;
    printf("Input i & j value: ");
    scanf("i=%d j=%d",&i,&j);
    printf("Your input was i=%d, j=%d\nInput i & j again: ",i,j);
    scanf("%d, %d",&i,&j);
    printf("Your input was i=%d, j=%d\n",i,j);
    printf("i + j = %d\n",i+j);
    return 0;
}
```

```
$ ./c06_8
Input i & j value: i=2 j=10
Your input was i=2, j=10
Input i & j again: 10, 11
Your input was i=10, j=11
i + j = 21
```

```
$ ./c06_8
Input i & j value: 10, 11
Your input was i=0, j=0
Input i & j again: Your input was
i=10, j=11
i + j = 21
```

```
$ ./c06_8
Input i & j value: 10 11
Your input was i=0, j=0
Input i & j again: Your input was
i=10, j=0
i + j = 10
```



C07.4 Конвертация из десятичной в восьмеричную систему счисления

c06_9-cvt2.c

```
#include <stdio.h>
// Convert demo program
int main()
{
    int i;
    printf("Input i: ");
    scanf("%d", &i);
    printf("Octal representation of i: %o\n", i);
    return 0;
}
```

```
$ ./c06_9-cvt2
Input i: 10
Octal representation of i: 12
```



C07.5 Распространенные ошибки при использовании scanf (передача переменной, а не указателя)

```
#include <stdio.h>
// Convert demo program
int main()
{
    int i = 0;
    printf("Input i: ");
    scanf("%d", i);
    printf("Octal representation of i: %o\n", i);
    return 0;
}
```

Правильно
scanf("%d", &i);

Компилятор способен обнаруживать подобные ошибки и сообщать о них:

```
$ gcc -Wall -o convert2 convert2.c
convert2.c: In function 'main':
convert2.c:7: warning: format '%d' expects type 'int *', but argument 2 has
type 'int'
```

**При выполнении такой программы она аварийно завершается,
не выводя на экран ожидаемую информацию!**



C07.6 Распространенные ошибки при использовании `scanf` (*разделитель в конце форматной строки*)

```
#include <stdio.h>
// Convert demo program
int main()
{
    int i = 0;
    printf("Input i: ");
    scanf("%d\n", &i);
    printf("Octal representation of i: %o\n", i);
    return 0;
}
```

Ошибка в том, что указан символ '\n' в конце форматной строки

При выполнении такой программы она запрашивает значение `i`, а потом "зависает"! На самом деле, согласно описанию функции `scanf`:

... символы разделители: пробел (' '), табуляция ('\t'), новая строка ('\n'), такой элемент соответствует любому количеству любых разделителей во входном тексте....

Таким образом, нажатие "Enter" совпадает с шаблоном и `scanf` продолжает считывать данные. Прервать работу `scanf` можно только введя символ-не-разделитель.



Домашняя работа

Разработанные программы необходимо написать и отладить. Финальный вариант переписывается от руки в тетрадь с домашними работами. Ответы на вопросы размещаются там же.



Н07.1 Модификация конвертера

Модифицировать программу `c06_9-cvt2.c` так, чтобы она выполняла перевод из десятичной в шестнадцатеричную систему счисления. Проверить ее работу на следующих входных данных:

- 1) 100
- 2) 128
- 3) 65280
- 4) 2650111222

Вопрос: Какие типы данных *достаточны*, если входные данные не превышают число 2)? 3)? 4)?



Н07.2 Модификация конвертера (2)

Модифицировать программу `c06_9-cvt2.c` так, чтобы она выполняла перевод из десятичной в шестнадцатеричную систему счисления. Проверить ее работу на следующих входных данных:

- 1) f5
- 2) ABFF
- 3) FEFAAA

Вопрос: Какие типы данных *достаточны*, если входные данные не превышают число 1)? 2)? 3)?



Н07.3 Действия над целыми числами

Задание

На вход программы поступает два целых числа, значения которых находятся в диапазоне $[-30000, 30000]$. Вычислить значения арифметических операций: $a + b$, $a - b$, $a * b$, a / b . Для $a = 10$ и $b = 5$ вывод программы должен выглядеть следующим образом:

$$10 + 5 = 15$$

$$10 - 5 = 5$$

$$10 * 5 = 50$$

$$10 / 5 = 2$$

Комментарии и вопросы:

1. В программе необходимо использовать наименьший по размеру допустимый тип данного. обоснуйте свой выбор.
2. Протестируйте программу на различных данных (не менее 10 наборов) и приведите результаты.
3. Объясните результаты, выдаваемые программой для чисел 10 и 3.
4. Для всех ли данных программа будет выдавать корректный результат? Если нет, то как это можно исправить?



Н07.4 Действия над вещественными числами

Задание

На вход программы поступает два вещественных числа, значения которых находятся в диапазоне $[-30000, 30000]$. Вычислить значения арифметических операций: $a + b$, $a - b$, $a * b$, a / b . Для $a = 10$ и $b = 3$ вывод программы должен выглядеть следующим образом:

$$\begin{array}{ll} 10 + 3 & = 13 \\ 10 - 3 & = 7 \\ 10 * 3 & = 30 \\ 10 / 3 & = 0.33 \end{array}$$

Комментарии и вопросы:

1. В программе необходимо использовать наименьший по размеру допустимый тип данного. обоснуйте свой выбор.
2. Протестируйте программу на различных данных (не менее 10 наборов) и приведите результаты.
3. Для всех ли данных программа будет выдавать корректный результат? Если нет, то как это можно исправить?



A07.1 Форматный вывод в командной строке

Команда **ifconfig** используется для конфигурирования сетевых интерфейсов ядра. Если аргументы не переданы, **ifconfig** выдает информацию о состоянии активных интерфейсов в следующем виде:

```
wlan0      Link encap:Ethernet  HWaddr AA:BB:CC:DD:EE:FF
            inet addr:192.168.0.208  Bcast:192.168.0.255  Mask:255.255.255.0
            inet6 addr: fe80::6e88:14ff:fe62:87cc/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:2539966 errors:0 dropped:0 overruns:0 frame:0
            TX packets:16547  errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:3367045445 (3.3 GB)  TX bytes:2146370 (2.1 MB)
```

Красным цветом выделены элементы выходной информации, используемые при выполнении данного задания:

MTU – Maximum Transfer Unit – максимальный размер передаваемого пакета.

RX/TX packets – количество принятых (RX – receive) и отправленных (TX – transmit) пакетов.

RX/TX bytes – общий размер принятой (RX) и отправленной (TX) информации в байтах.



A07.1 Форматный вывод в командной строке

Задание:

Разработайте и отладьте программу a07_1.c, на вход которой поступает:

1. Максимальный размер пакета в сети (MTU).
2. Количество переданных пакетов (TX packets).
3. Количество принятых пакетов (RX packets).

Вывести информацию о сетевом интерфейсе wlan0, так как она приведена на слайде, заменив элементы, выделенные красным цветом, согласно введенной информации. При расчете RX/TX bytes, считать, что принимались и передавались только пакеты максимально возможного размера.

```
wlan0      Link encap:Ethernet  HWaddr AA:BB:CC:DD:EE:FF
            inet addr:192.168.0.208  Bcast:192.168.0.255  Mask:255.255.255.0
            inet6 addr: fe80::6e88:14ff:fe62:87cc/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:2539966 errors:0 dropped:0 overruns:0 frame:0
            TX packets:16547  errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:3367045445 (3.3 GB)  TX bytes:2146370 (2.1 MB)
```