

# Задание №3.3 в рамках вычислительного практикума. Представление в памяти строк и массивов строк

## Строка

Исходный код с использованием строки, представленной в виде массива символов:

```
int main(void)
{
    char str[] = "Hello, world!";
    return 0;
}
```

Дамп всей строки:

```
(gdb) x/14xb str
0x5ffe92:  0x48  0x65  0x6c  0x6c  0x6f  0x2c  0x20  0x57
0x5ffe9a:  0x6f  0x72  0x6c  0x64  0x21  0x00
```

Объяснение: каждый символ представлен размером в 1 байт. 0x48 – соответствует символу “H” в таблице ASCII, 0x65 – соответствует символу “e”, и так далее. 0x00 - этот байт представляет нулевой символ (\0), который указывает на конец строки.

## Массив строк

Исходный код с использованием массива строк

```
int main(void)
{
    char words[][10] = { "apple", "banana", "pineapple" };
    return 0;
}
```

Дамп всего массива:

(gdb) x/30xb words

|           |      |      |      |      |      |      |      |      |
|-----------|------|------|------|------|------|------|------|------|
| 0x5ffe80: | 0x61 | 0x70 | 0x70 | 0x6c | 0x65 | 0x00 | 0x00 | 0x00 |
| 0x5ffe88: | 0x00 | 0x00 | 0x62 | 0x61 | 0x6e | 0x61 | 0x6e | 0x61 |
| 0x5ffe90: | 0x00 | 0x00 | 0x00 | 0x00 | 0x70 | 0x69 | 0x6e | 0x65 |
| 0x5ffe98: | 0x61 | 0x70 | 0x70 | 0x6c | 0x65 | 0x00 |      |      |

Слово `apple` состоит из 5 букв, значит полезными данными в памяти будут первые 6 байт, с учетом нулевого символа. Оставшиеся 4 байта – вспомогательные данные.

Слово `banana` состоит из 6 букв, значит полезными данными в памяти будут байты с 11 по 17, с учетом нулевого символа. Оставшиеся 3 байта – вспомогательные данные.

Слово `pineapple` имеет 9 букв, значит полезными данными в памяти будут байты, с 21 по 29. Оставшийся 1 байта – вспомогательные данные.

Итого:

- Размер “полезных” данных:  $6 + 7 + 10 = 23$  байта
- Размер “вспомогательных” данных:  $4 + 3 + 0 = 7$  байт
- % «вспомогательных» данных по отношению ко всем данным: 23.3%

## Массив строк, как массив указателей

Исходный код с использованием массива строк, как массива указателей

```
int main(void)
{
    char *words[] = { "apple", "banana", "pineapple" };
    return 0;
}
```

Дамп всего массива указателей:

(gdb) x/24xb words

|           |      |      |      |      |      |      |      |      |
|-----------|------|------|------|------|------|------|------|------|
| 0x5ffe80: | 0x00 | 0x40 | 0x47 | 0x85 | 0xf7 | 0x7f | 0x00 | 0x00 |
| 0x5ffe88: | 0x06 | 0x40 | 0x47 | 0x85 | 0xf7 | 0x7f | 0x00 | 0x00 |
| 0x5ffe90: | 0x0d | 0x40 | 0x47 | 0x85 | 0xf7 | 0x7f | 0x00 | 0x00 |

Размер вспомогательных данных:  $3 * 8 = 24$  байта

Дамп первой строки:

```
(gdb) x/6xb words[0]  
0x7ff785474000: 0x61  0x70  0x70  0x6c  0x65  0x00
```

Размер полезных данных: 6

Дамп второй строки:

```
(gdb) x/7xb words[1]  
0x7ff785474006: 0x62  0x61  0x6e  0x61  0x6e  0x61  0x00
```

Размер полезных данных: 7

Дамп третьей строки:

```
(gdb) x/10xb words[2]  
0x7ff78547400d: 0x70  0x69  0x6e  0x65  0x61  0x70  0x70  0x6c  
0x7ff785474015: 0x65  0x00
```

Размер полезных данных: 10

Общий размер полезных данных:  $6 + 7 + 10 = 23$  байта

% «вспомогательных» данных по отношению ко всем данным: 48.9%