



Universidade do Minho
Escola de Engenharia

MIETI :: Métodos de Programação II
2015/16

Práticas Laboratoriais

Módulo 2

António Esteves
esteves@di.uminho.pt
Março 2016

Exercício 1 - Apontadores

- ◇ Desenhe uma tabela para representar a memória do computador quando está a executar um programa:

- ◇ Assuma que cada linha na tabela guarda uma palavra (um int de 32-bits ou um apontador).
- ◇ Rotule as linhas com endereços a começar em **100** e a crescer de cima para baixo:

100	
104	
108	
112	
116	
120	
...	

Exercício 1 - Apontadores

- ◇ Atribua um endereço de memória a cada variável do programa que estamos a executar.
- ◇ Por exemplo, podemos considerar que a variável **char *s** é armazenada a partir da posição de memória **104** e a variável **int i** é armazenada a partir da posição de memória **112**:

100	
104	variável s
108	
112	variável i
116	
120	
...	

Exercício 1 - Apontadores

- ◇ Suponha ainda que o programa em execução usa uma *string* de caracteres, com o seguinte conteúdo:

s = "Hi Fred";

- ◇ Isto significa que a *string* de caracteres "Hi Fred" está guardada algures na memória, e que o valor de **s** é o endereço do primeiro byte da *string*. Vamos assumir que o compilador coloca a *string* (apontada por **s**) em memória a partir do endereço **116**.
- ◇ Vamos ainda assumir que atribuímos um valor a **i**, por exemplo:

i = 22;

- ◇ A figura a seguir apresenta parte do conteúdo das posições de memória utilizadas. Preencha o conteúdo das duas linhas em falta.

Exercício 1 - Apontadores

100				
104	valor de s =			
108				
112	valor de i =			
116	H	i		F
120	r	e	d	\0
...				

Exercício 1 - Apontadores

- ◇ Complete a seguinte tabela, onde se apresentam vários exemplos de expressões em C relativos à figura anterior e ao que foi sendo descrito no texto:

Expressão	Descrição do que representa a expressão	Valor (com base no texto)
<code>i</code>		
<code>&i</code>		
<code>s</code>		
<code>&s</code>		
<code>*s</code>		
<code>s[3]</code>		
<code>s+2</code>		
<code>*(s+2)</code>		

Exercício 1 - Apontadores

- ◇ Quer saber um pouco mais sobre apontadores? Então considere que o programa inclui a seguinte instrução:

char **p = &s;

- ◇ Agora, a variável **p** contém o endereço de **s**. Suponha que **p** é guardada a partir do endereço de memória **100**. Preencha o conteúdo das 3 linhas em falta:

100	valor de p =			
104	valor de s =			
108				
112	valor de i =			
116	H	i		F
120	r	e	d	\0
...				

Exercício 1 - Apontadores

- ◇ Complete a seguinte tabela, onde se apresentam alguns exemplos de expressões em C relativos à figura e texto anteriores:

Expressão	Descrição do que representa a expressão	Valor (com base no texto)
p		
&p		
*p		
*(p)		

Exercício 2 – Array bidimensional

- ◇ Implemente e teste um programa em C que conta e apresenta no ecrã o número de ocorrências de cada letra do alfabeto ('a' a 'z') presente num *array* de caracteres bidimensional `m[5][10]`. Teste o programa inicializando o *array* `m[][]` com os seguintes valores:

c	x	a	g	R	t	z	q	o	g
f	e	u	c	r	y	u	i	l	M
a	h	k	L	v	p	t	n	B	s
y	J	x	m	o	f	z	a	u	r
z	T	q	b	g	C	e	y	R	q

- ◇ Na solução, não diferencie as letras minúsculas das maiúsculas. Por exemplo, considere 'c' e 'C' como sendo a mesma letra. Deste modo o programa deve indicar a existência de 3 ocorrências de 'c' no *array* `m[][]`: 'c', 'c', e 'C'.

Exercício 2 – Array bidimensional

- ◇ Para resolver o problema implemente e utilize as seguintes funções:

```
void contarLetras(char m[][10], int nLin, int nCol, int cnt[], int nL);
```

Função que conta o número de ocorrências de cada letra no *array* `m`.

```
void printNumeroLetras(int cnt[], int nL);
```

Função que escreve no ecrã o número de ocorrências de cada letra, contido no *array* `cnt`.

`m` → *array* de caracteres declarado no `main()`

`nLin` → número de linhas do *array* `m`

`nCol` → número de colunas do *array* `m`

`cnt` → *array* onde se guarda o número de ocorrências de cada letra

`nL` → número de letras do alfabeto.

Exercício 3 – Apontadores para funções

- ◇ Escrever um programa que pede ao utilizador o que quer calcular:
 - Perímetro duma circunferência de raio $r \rightarrow p = 2 \cdot \text{PI} \cdot r$
 - Área dum círculo de raio $r \rightarrow A = \text{PI} \cdot r^2$
 - Volume duma esfera de raio $r \rightarrow V = \frac{4}{3} \cdot \text{PI} \cdot r^3$
- ◇ Para isso deve usar 3 funções, uma para cada cálculo, que recebem como argumento um `float` (raio) e não devolvem nada.
- ◇ O programa deve usar um *array* de apontadores para estas 3 funções.
- ◇ De acordo com a escolha do utilizador, chama-se a função adequada desse *array*.
- ◇ Cada função recebe como argumento o raio, faz os cálculos e escreve o resultado no ecrã.
- ◇ Para resolver este problema utilize como ponto de partida o exercício apresentado nas aulas teóricas.