

**Aula 1, segunda-feira, 15 março, 3h****Aula 2, quarta-feira, 17 março, 1h30**

Apresentação; Introdução à programação em C; I/O

Arrays; Strings; Funções

<p>Apresentação da unidade curricular: enquadramento; programa; bibliografia; funcionamento; avaliação.</p> <p>Introdução ao Unix: aspetos básicos; interpretador de comandos e comandos comuns.</p> <p>Aspetos conceituais do sistema Unix: redirecionamento de entradas e saídas; encadeamento de comandos.</p> <p>Introdução à linguagem C: estrutura básica de um programa; tipos numéricos char e int.</p> <p>Funções de entrada e saída getchar, putchar e printf.</p> <p>Operações de edição de programas; geração de programa executável; execução de programas; <i>debugging</i> de programas.</p> <p>Exemplo: helloworld; contar caracteres; contar linhas; contar palavras.</p> <p>Bibliografia: The C Programming Language, cap1, secções 1.1 a 1.6</p>	<p>Array - agregado de elementos do mesmo tipo; indexação; dimensão.</p> <p>Exemplo: contador de palavras por dimensão das palavras.</p> <p>Funções - passagem de argumentos por valor; arrays como argumentos.</p> <p>Operador sizeof.</p> <p>Exemplo: ordenação de array de inteiros com recurso a função.</p> <p>Strings - caso particular de array de caracteres, com valor terminador definido por convenção.</p> <p>Funções de biblioteca para manipulação de strings.</p> <p>Exemplo: função para leitura de uma linha de texto; funções para manipulação de strings C.</p> <p>Bibliografia: <i>The C Programming Language</i>, cap. 1, secções 1.5 a 1.10</p>
---	---

**Aula 3, segunda-feira, 22 março, 3h****Aula 4, quarta-feira, 24 março, 1h30**

Tipos, Operandos e Expressões

Ponteiros e arrays

<p>Tipos numéricos em C: tipos numéricos inteiros e reais; portabilidade; avaliação booleana; conversão entre tipos.</p> <p>Operações <i>bit-a-bit</i>.</p> <p>Precedência e associatividade dos operadores.</p> <p>Exemplos: <i>getbits</i>(para obter bits de um inteiro); <i>packdate</i> (representação de datas sobre inteiros a 16 bits).</p> <p>Bibliografia: <i>The C Programming Language</i>, cap. 2; Computer Systems, cap 2.</p>	<p>Ponteiros enquanto referências explícitas. Operadores * (conteúdo de) e &amp; (endereço de).</p> <p>Exemplo: função <i>date_unpack</i>(short, int*year, int* month, int* day).</p> <p>Relação entre ponteiros e arrays.</p> <p>Ponteiros para caracteres.</p> <p>Exemplo: ordenação de array de valores inteiros com função <i>swap</i>; funções para processamento de strings.</p> <p>Bibliografia: <i>The C Programming Language</i>, cap. 5, secções 1 a 5</p>
--	--

**segunda-feira, 29 março****quarta-feira, 31 março**

Páscoa

Páscoa

--	--

	<p>Aritmética de ponteiros.</p> <p>Array multidimensional.</p> <p>Array de ponteiros.</p> <p>Ponteiros para ponteiros.</p> <p>Argumentos na linha de comando.</p> <p>Exemplo: ordenação de lista de nomes na forma de array bidimensional de caracteres ou na forma de array de ponteiros para strings.</p> <p>Bibliografia: <i>The C Programming Language</i>, cap. 5, secções 6 a 10</p>
--	--

<p>Tipos agregados - struct: sequências de valores, de tipos diferentes, contíguos em memória.</p> <p>Passagem de parâmetros e retorno de valores do tipo struct. Ponteiros para struct. Aritmética de ponteiros.</p> <p>Alinhamento em memória de variáveis e de membros de estruturas. Operador sizeof aplicado a structs.</p> <p>Exercício:</p> <p>Programa contador de ocorrências de palavras tendo como estrutura de dados base um array de struct.</p> <p>Tipo union - interpretação do mesmo conteúdo de memória na perspetiva de diferentes tipos de dados.</p> <p>Representação de valores reais em vírgula flutuante segundo a norma IEEE 754 para precisão simples.</p> <p>Exercício:</p> <p>Decomposição da representação de um valor do tipo float nas suas componentes.</p> <p>Bibliografia: <i>The C Programming Language</i>, cap. 6, secções 1 a 4 e 8</p>	<p>Modelo de ficheiro de texto.</p> <p>Funções da biblioteca para acesso a ficheiros.</p> <p>Exercícios:</p> <p>Cópia de ficheiros. Concatenação de ficheiros.</p> <p>Funções com número de parâmetros variável.</p> <p>Exercício:</p> <p>Função para adição de um número variável de parâmetros.</p> <p>Bibliografia: <i>The C Programming Language</i>, cap. 6, secções 1 a 4 e 8</p>
--	---

**Aula 8, segunda-feira, 19 abril, 3h****Aula 9, quarta-feira, 21 abril, 1h30**

## Introdução à arquitetura x86\_64

## x86\_64 – Acesso a dados

<p>Introdução da arquitetura x86_64.</p> <p>Representação dos tipos de dados – ordem e alinhamento.</p> <p>Conjunto de registos.</p> <p>Conjunto de instruções – caracterização por grupos e definição dos parâmetros.</p> <p>Mecanismo de chamada a função.</p> <p>Exercício: análise de programa em <i>assembly</i>, resultado da compilação de um programa em C.</p> <p>Código padrão das principais estruturas de controlo das linguagens de alto nível (if / while / for).</p> <p>Linguagem assembly – sintaxe, diretivas, ferramentas.</p> <p>Exercício: tradução para assembly de uma implementação de strlen em linguagem C.</p> <p>Bibliografia: Computer Systems - A Programmer's Perspective, 3rd ed, secções 3.3 e 3.4.</p>	<p>Acesso a dados em array de tipos simples, a campos de struct e array de struct.</p> <p>Alinhamento.</p> <p>Exercícios:</p> <ul style="list-style-type: none"><li>• Procurar o maior valor num array de inteiros.</li><li>• Procurar a pessoa mais alta num array de pessoas.</li></ul> <p>Bibliografia: Computer Systems - A Programmer's Perspective, 3rd ed, secções 3.6 a 3.9</p>
---	---

**Aula 10, segunda-feira, 26 abril, 3h****Aula 11, quarta-feira, 28 abril, 1h30**

## x86\_64 - Convenções de chamada a funções

Exercícios de programação em *assembly*

<p>Codificação de funções em linguagem <i>assembly</i> para interoperação com a linguagem C. Representação de dados, passagem de parâmetros e retorno de valores, critérios de utilização dos registos (caller saved; callee saved).</p> <p>Exercício: programação em assembly de uma função de ordenação de um array de inteiros com recurso a função auxiliar para troca de elementos do array.</p> <p>Bibliografia: Computer Systems - A Programmer's Perspective, 3rd ed, secções 3.7</p>	<p>Codificação de função folha para acesso a estrutura de dados complexa.</p>
---	---

Funções genéricas, ponteiros genéricos e ponteiros para função	x86_64 - Stack frame
<p>Ponteiros genéricos com void *.</p> <p>Funções genéricas - descrição genérica de <i>arrays</i> com três argumentos; ponteiros para função.</p> <p>Funções genéricas presentes na biblioteca <i>standard</i> do C.</p> <p>Escrita de função genérica para ordenação de elementos num <i>array</i>.</p> <p>Aplicação de funções genéricas na ordenação de <i>arrays</i> de diversos tipos de elementos – tipos simples, ponteiros ou struct.</p> <p>Exercício: programação de algoritmo <i>bubble_sort</i> genérico. Utilização da função <i>qsort</i> na ordenação de array de struct.</p> <p>Bibliografia: The C Programming Language, secção 5.11.</p>	<p>Stack frame - organização do stack para suporte à execução: zona de argumentos; endereço de retorno; área para salvaguarda de registos; área reservada às variáveis locais; área reservada à passagem de argumentos na chamada a outras funções.</p> <p>Análise da execução de uma sequência de chamadas envolvendo a passagem de parâmetros em stack e a instanciação de variáveis locais em stack.</p> <p>Exercício: programação em assembly de sequência de chamadas a funções com mais de 6 parâmetros e aplicação de operador &amp; a variáveis locais e parâmetros.</p> <p>Bibliografia: Computer Systems - A Programmer's Perspective, 3rd ed, secções 3.10.</p>

## x86\_64 – Buffer overflow

## Alocação dinâmica de memória

<p>Acesso a variáveis para além dos seus limites – causas e efeitos.</p> <p>Ataque por <i>buffer overflow</i>.</p> <p>Defesas – utilização de espaços de endereçamento aleatórios, inibição de execução de código em <i>stack</i>, <i>stack clash protector</i>; <i>stack corruption detection</i></p> <p>Exercício – análise de código vulnerável (<i>secrets.c</i>)</p> <p>Bibliografia: Computer Systems - A Programmer's Perspective, 3rd ed, secção 3.10.</p>	<p>Alojamento de variáveis (objetos) – estático; registo; stack; dinâmico (implícito /explícito)</p> <p>Introdução às estruturas de dados dinâmicas.</p> <p>Interface de utilização disponível na biblioteca normalizada da linguagem C: <i>malloc</i>, <i>calloc</i>, <i>realloc</i> e <i>free</i>.</p> <p>Exercício: programa de simulação de uma fila de espera.</p> <p>Bibliografia: The C Programming Language, secção 6.5</p>
--	---

**Aula 16, segunda-feira, 17 maio, 3h****Aula 18, quarta-feira, 19 maio, 1h30**

## Estruturas de dados dinâmicas I

## Estruturas de dados dinâmicas II

<p>Implementação de lista ligada genérica não intrusiva.</p> <p>Exercício: adaptação do programa de simulação de fila de espera à utilização de uma lista ligada genérica, não intrusiva.</p> <p>Bibliografia: <i>The C Programming Language</i>, secções 6.5 a 6.7 e secção 4.11</p>	<p>Implementação de lista ligada genérica intrusiva – macro <code>offsetof</code>.</p> <p>Exercício: adaptação do programa de simulação de fila de espera à utilização de uma lista ligada genérica, intrusiva.</p> <p>Bibliografia: <i>The C Programming Language</i>, secções 6.5 a 6.7 e secção 4.11</p>
---	---

**Aula 19, segunda-feira, 24 maio, 3h****Aula 21, quarta-feira, 26 maio, 1h30**

## Gestão dinâmica de memória

## Estruturas de dados dinâmicas III

<p>Análise da implementação de gestão de memória apresentada no livro “C Programming Language.</p> <p>Implementação da função <code>realloc</code> sobre o gestor de memória estudado.</p>	<p><i>Array</i> dinâmico.</p> <p>Exercício – Contador de palavras com base em lista ligada e em <i>array</i> dinâmico.</p> <p>Medida de tempo de execução.</p>
--	--

**Aula 22, segunda-feira, 31 maio, 3h****Aula 23, quarta-feira, 2 junho, 1h30**

## Bibliotecas

## Compilação separada

<p>Introdução à compilação separada – <code>makefile</code>.</p> <p>Bibliotecas estáticas – criação e utilização.</p> <p>Exercício: criação de biblioteca com a implementação dos contentores realizados em aulas anteriores – lista ligada, árvore binária, <i>array</i> dinâmico e tabela de <i>hash</i>.</p> <p>Bibliografia: <i>Computer Systems: a programmer's perspective</i>, cap. 7, secções 5 a 7;</p> <p>Preparação para a realização da 3ª série de exercícios:</p> <p>Introdução ao protocolo HTTP e à norma de formatação de dados JSON.</p> <p>Realização de programa de exemplo para acesso a recursos na Internet através do protocolo HTTP e utilizando a biblioteca <code>CURL</code>.</p> <p>Realização de programa exemplo para extração de informação representada em formato JSON, utilizando a biblioteca <code>Jansson</code>.</p>	<p>Modelo de compilação separada.</p> <p>Processo completo de produção de um programa: pré-processamento, compilação e ligação. Declaração e definição. Visibilidade/âmbito dos símbolos.</p> <p>Secções principais dos ficheiros objeto relocizáveis: <code>.text</code>, <code>.rodata</code>, <code>.data</code>, e <code>.bss</code>.</p> <p>Ligação: resolução de símbolos (símbolos não definidos e símbolos duplicados); relocização.</p> <p>Exercício: análise do processo de compilação e ligação utilizando os programas utilitários <code>objdump</code>, <code>readelf</code> e <code>nm</code>. Utilização dos programas <code>main.c</code> e <code>add.c</code>.</p> <p>Bibliografia:</p> <p><i>Computer Systems: a programmer's perspective</i>, cap. 7, secção 1 a 7</p> <p><i>The C Programming Language</i>, cap. 4, secções 3 a 5</p>
---	---

**Aula 24, segunda-feira, 7 junho, 3h****Aula 25, quarta-feira, 9 junho, 1h30**

## Bibliotecas de ligação dinâmica

## Código independente de posição

<p>Revisão do processo de realocização realizado em compile time.</p> <p>Ficheiro objeto executável – composição; carregamento em memória.</p> <p>Bibliotecas de ligação dinâmica (shared objects) - contextualização, criação e utilização.</p> <p>Código independente de posição.</p> <p>Exercício: criação de ficheiros makefile para criação e utilização de biblioteca de ligação dinâmica.</p> <p>Bibliografia: Computer Systems: a programmer's perspective, cap. 7, secções 7 a 10;</p>	<p>Análise do código de referência a funções e variáveis entre o programa executável e a biblioteca de ligação dinâmica.</p> <p>Bibliografia: Computer Systems: a programmer's perspective, cap. 7, secção 12;</p>
---	--

**Aula 26, segunda-feira, 14 junho, 3h****Aula 27, quarta-feira, 16 junho, 1h30**

## Carregamento Dinâmico de Bibliotecas

## Hierarquia de memória - cache

<p>Carregamento explícito de bibliotecas de ligação dinâmica em tempo de execução. As operações dlopen, dlsym e dlclose.</p> <p>Exemplo: realização de plug-in para o programa de simulação de filas de espera.</p> <p>Resolução de exercícios de testes passados.</p>	<p>Enquadramento da memória <i>cache</i> na hierarquia de memória.</p> <p>Localidade temporal e localidade espacial.</p> <p>Arquitetura da <i>cache</i>.</p>
--	--

**Aula 28, segunda-feira, 21 junho, 3h****Aula 29, quarta-feira, 23 junho, 1h30**

## Hierarquia de memória - cache

<p>Exercícios de medida de desempenho de programas com vista a evidenciar o efeito da <i>cache</i>.</p>	
---	--