

1 Apresentação e objectivos

Introdução à Programação

Estes apontamentos das aulas teóricas são da autoria de [Pedro Vasconcelos](#) (2007) tendo sido adaptados e modificados por [Armando Matos](#) (2010)

Introdução à Programação Objectivos

1. Introdução às linguagens de programação
2. Análise, estruturação, codificação e teste de algoritmos simples na linguagem de programação `python`
3. Conhecimento de alguns algoritmos básicos (pesquisa, ordenação...)

Introdução à Programação

Docente: Armando Matos `acm@dcc.fc.up.pt`

Bibliografia recomendada

Referências

- [1] “*How to Think Like a Computer Scientist*”, A. Downey, J. Elkner e C. Meyers. Green Tea Press. PDF disponível em <http://www.greenteapress.com/thinkpython/thinkCSpy/thinkCSpy.pdf>.
- [2] “*Problem Solving with Algorithms and Data Structures*”, B. Miller and D. Ranum. Franklin, Beedle & Associates, 2006.

2 Introdução à Programação

Introdução à Programação

- Computador: processador de *informação* (dados) segundo uma lista de *instruções* (programa) [Estruturas de dados](#) ↔ [Algoritmos](#)
- efectua operações simples (aritméticas/lógicas) depressa:
humano: aproximadamente 1 operação/segundo
computador actual: >1 milhão de operações/segundo
- qualquer tipo de dados *quantificáveis* (e.g. números, textos, sons, imagens...)
- universal: utilizável para diversos fins com diferentes *programas*
- Modelo de execução de von-Neumann

Breve cronologia dos computadores

1940s	Colossus, Harvard Mk I, ENIAC	2ª Guerra Mundial
1950s	UNIVAC I	computadores comerciais
1960s	IBM System/360	
1970s	PDP-11, DEC VAX, VMS, UNIX	
1977	Apple II	computadores pessoais
1981	IBM PC	
1984	Apple Mac	interfaces gráficas
1986	Intel 386, Windows 1.0	
1990	Windows 3.0	
1991	WWW, GNU/Linux	internet, open-source
1993	Intel Pentium	
1995	Windows 95	
2001	Windows XP, MacOS X	
2001	Windows Vista	

Níveis conceptuais de um computador

Hardware: CPU, memória, unidades de disco, ecrã, teclado, rato. . .

Software: sistema operativo, aplicações, jogos, ficheiros de dados (imagens, músicas, filmes, folhas de cálculo, bases de dados. . .)

Tendências:

- *hardware* mais barato
- *software* cada vez mais caro (percentualmente)
- desenvolvimento de *software* mais rápido:
 - linguagens de alto nível
 - re-utilização de componentes (bibliotecas)

Sistemas de operação

- Conjunto de *software* de base para gerir recursos do computador
- Proporciona *funcionalidades* para as aplicações:
 - gestão de utilizadores
 - gestão de memória
 - gestão de ficheiros
 - gestão de I/O: terminais, impressoras, interfaces gráficas, ligações de rede

Cronologia do sistemas de operação (1)

Primeiros sistemas de operação (1950s):

- um trabalho de cada vez (*batch*)
- apenas supervisiona as transições entre trabalhos

Mainframes (1960-1970s)

- caros: necessário partilhar recursos
- multi-utilizador: vários utilizadores em *terminais*
- multi-tarefa: divisão o tempo de processamento entre os vários trabalhos (*time-sharing*)

Cronologia dos sistemas de operação (2)

Mini-computadores: sistema UNIX (1970s)

- multi-utilizador, multi-tarefa
- portátil para diferentes modelos de computadores
- “código-fonte” em linguagem C distribuído com o sistema
- popular na comunidade académica (Universidade de Berkeley)
- variantes comerciais: Ultrix, System V, IRIX, Solaris

Cronologia dos sistemas de operação (3)

Primeira geração de computadores pessoais (1970–1980)

- recursos escassos: um utilizador, uma tarefa
- interface textual (e.g. interpretador de comandos MS-DOS)
- combinado com uma linguagem de programação (BASIC)
- específicos de cada modelo de computador e.g. Apple II, IBM PC

Cronologia dos sistemas de operação (4)

Primeiras interfaces gráficas (1984–1990)

- aplicações associadas a *janelas* separadas
- apresentação de programas e ficheiros usando *ícones*
- selecção visual usando o “rato”
- facilita o acesso aos utilizadores
- um utilizador, multi-tarefa
- associadas a um modelo de *hardware* específico

Cronologia dos sistemas de operação (5)

Actualmente:

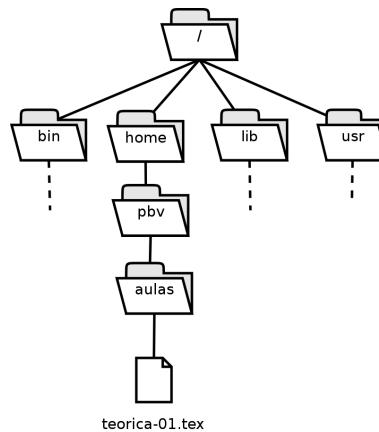
- computadores pessoais têm recursos do que grandes computadores de gerações anteriores
- ligados em *redes* locais e globais (internet)
- multi-utilizador, multi-tarefa
- maior independência do *hardware*
- separação entre o *núcleo* e a *interface gráfica*
- UNIX em PCs: GNU/Linux, Free BSD, MacOS X

O sistema UNIX

- organizado num *núcleo (kernel)* e vários *processos*
 - núcleo:** tem acesso directo ao *hardware*
 - processos:** pedem recursos ao núcleo (e.g. consultar ficheiros)
- cada utilizador:
 - identificado por um “*login*” e.g. pbv
 - autenticado por uma *palavra-passe (secreta)*
 - área pessoal para ficheiros: “*home directory*”
- um utilizador especial: `root`
 - único que pode alterar configurações de sistema
 - único que pode acrescentar/remover utilizadores

Organização de ficheiros

- *ficheiros*: textos, imagens, programas...
- identificados por *nomes*
- estruturados em *directórios* hierárquicos e.g. `/home/pbv/aulas/teorica-01.tex`
- *permissões* associadas a cada ficheiro: leitura, escrita, execução



Processos

- processo: execução dum programa num determinado contexto (utilizador e dados)
- aparência de vários processos a “correr” em simultâneo
- processos de utilizadores comuns: aplicações, editores, compiladores, interpretadores, etc.
- processos de sistema: interface gráfica, servidores (WWW, email, ssh), etc.
- filosofia UNIX: núcleo simples, tudo o resto são processos!

Interpretador de comandos UNIX

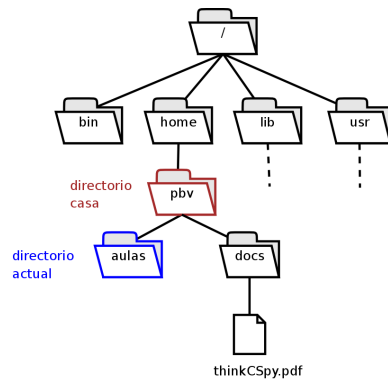
- “*shell*”: interface textual para executar comandos UNIX
- cada comando é (normalmente) um programa em `/bin` ou `/usr/bin`
- sintaxe típica: `comando [opções] [arg1 arg2 ...]`
- cada comando é executado com um novo processo
- associado ao *directório corrente* (“*working directory*”)

Alguns comandos úteis

ls	listar ficheiros no directório actual
pwd	imprime o directório actual
cd	mudar o directório actual
mkdir	criar um novo directorio
rmdir	remover um directório (vazio)
cp	copiar ficheiro
mv	mover/mudar nome de um ficheiro
rm	remover um ficheiro
cat	listar o conteúdo de ficheiros (more)
ps	listar processos (do utilizador ou do sistema)
man	mostrar manual de um comando (info)

Caminhos absolutos e relativos

.	directório actual
..	directório pai
~	directório casa



- absoluto: `/home/pbv/aulas/docs/thinkCSpy.pdf`
- relativo: `../docs/thinkCSpy.pdf`
- relativo à casa: `~/docs/thinkCSpy.pdf`

Na próxima aula

- Linguagens de programação
- Apresentação da linguagem *Python*