

Departamento de Ciência de Computadores
 Faculdade de Ciências
 Universidade do Porto

Miguel Filgueiras, Sumários de “Introdução aos Computadores”

Introdução aos Computadores
Introdução à Ciência de Computadores
Sumários—2010/11

22/9 Objectivos e funcionamento da disciplina.

Sistemas Unix e Linux. Sistemas de janelas, janelas tipo terminal, “shell” (casca), execução de programas e comandos. O editor **emacs** e como consultar o tutorial. Sistema de ficheiros: ficheiros ordinários, directórios, ligações, e ficheiros especiais; árvore de ficheiros e directórios habituais em Linux; directório casa e directório corrente. Caminhos para ficheiros: caminho absoluto e relativo. Alguns comandos relacionados com ficheiros: **ls**, **pwd**, **cd**. Opção **-l** do comando **ls**, permissões de acesso de ficheiros, proprietário, grupo, execução de programas e mudança de directório. Comando para consultar o manual (**man**).

27/9 Os ficheiros **.** e **..** em directórios. Abreviaturas para nomes de ficheiros: **~**, *****, **?**, conjuntos de símbolos. Comandos para gestão de ficheiros (**cp**, **mv**, **rm**, **mkdir**). Alteração de permissões de acesso, proprietário e grupo (**chmod**, **chown**, **chgrp**).

TP, 27/9 e 29/9 Edição na linha de comando da “shell”. História de comandos.

Entrada/saída de programas, canais standard, redirecção e canos (“pipe”).

Utilitários para ficheiros: des/compressão (**gzip**, **bzip2**, **zip**), arquivos (**tar**), diferenças (**diff**, **tkdiff**).

P, semana de 27/9 Gerir ficheiros (**ls**, **cp**, **mv**, **rm**, **mkdir**). Arquivos (**tar**). Editor de texto: **emacs**.

29/9 Representação binária de informação: bases de numeração, números binários, octais e hexadecimais.

Representação de números em computador: inteiros não negativos, inteiros negativos, representação em complemento para 2, representação deslocada de N , números reais, representação em vírgula flutuante, norma IEEE 754.

Digitalização de informação: informação discreta e contínua, textos, sons e imagens, perdas por discretização e por limite no tamanho da representação.

4/10 Noções sobre arquitectura e funcionamento de computadores: equipamento (“hardware”), software, informação. Estrutura física de computadores actuais: placa-mãe (“motherboard”), relógio, barramento(s) (“bus”), processador, memória, interfaces, fonte de alimentação.

Circuitos lógicos: transistores; modelos abstractos.

TP, semana de 4/10 Representação de imagens a 2 cores como mapa de bits. Operações lógicas: negação, conjunção, disjunção (inclusiva), disjunção exclusiva, equivalência, implicação, negação da conjunção (“nand”) e negação da disjunção (“nor”). Máscaras para extracção de informação empacotada, deslocamentos e rotações.

P, semana de 4/10 e dia 12/10 Conversão de base de números: bases 2 (usando uma tabela de potências de 2), 8 e 16. Representação de números negativos em complemento para 2: cálculos. Representação de números em vírgula flutuante.

Departamento de Ciência de Computadores

Faculdade de Ciências

Universidade do Porto

Miguel Filgueiras, Sumários de “Introdução aos Computadores”

6/10 Modelos abstractos de circuitos lógicos. Redes neuronais de McCulloch-Pitts: células, entradas de activação e de inibição, saída, limiar, atraso, estados das linhas. Células para operações lógicas básicas.

Redes neuronais para porta bidireccional, soma, e memória de 1 bit (sem descodificação da linha de dados). Circuitos electrónicos de memória: SR-flip-flop e *D-type transparent latch*.

11/10 Diagrama de blocos de uma máquina de calcular usando portas, somador e memórias. Execução de operações a partir de teclas. Generalização da máquina para inteiros com n bits. Generalização da máquina para haver mais operações aritméticas: códigos de teclas, memória da operação, descodificador. Generalização da máquina para poder ser programada: estados e transição entre estados, programas, programa em memória, registo de sequência (“program counter”), ciclo de execução, descodificação de instruções.

TP, semana de 11/10 Representação de cores por RGB. Campos de bits, construção de máscaras, desempacotamento e empacotamento de informação.

P, semana de 11/10 e dia 19/10 `tkgate`: tutoriais *Creating a circuit, Combinational simulation*. Análise do comportamento de um SR-flip-flop.

18/10 Máquinas programáveis e máquinas com programa fixo: noções de complexidade e computabilidade, modelos de computação. Autómatos finitos e tradutores finitos: estados, transições, símbolos de entrada e de saída, estados inicial e finais. Modelos de controladores de semáforos de trânsito e de um elevador. Reconhecimento de linguagens e autómatos finitos. Exemplos.

TP, semana de 18/10 Ordenação de informação guardada em mais de uma posição de memória (“endianness”). Alinhamento de informação na memória. Resolução de problemas sobre colocação na memória de valores numéricos de acordo com a ordenação.

P, semana de 18/10 e dia 26/10 Desenho de redes neuronais para operações e expressões lógicas e para operações sobre números binários. Diagramas com a temporização de sinais.

20/10 Reconhecimento de linguagens e autómatos finitos (cont.): programas baseados em autómatos.

Modelos para uma máquina que faz trocos e para um reconhecedor de múltiplos de 3. Controle de máquinas de estados: micro-controladores, entrada/saída, leitura de teclados, relés. Análise de um modelo físico de um reconhecedor de múltiplos de 3. Referência a sistemas embebidos (“embedded systems”) e sistemas de tempo real.

25/10 Computadores e programação-máquina. Instruções-máquina, linguagem-máquina, ensambladores. Programação com uma linguagem-máquina mínima para inteiros sem sinal: instruções para incrementar, colocar em zero, e transferir se zero ou decrementar. Pseudo-instruções para: transferência incondicional, paragem, mover (copiar), trocar, decrementar, somar, subtrair, diferença em valor absoluto, multiplicação, transferências condicionais ($= 0$, $\neq 0$, $<$, \geq , $=$), e divisão inteira.

Computadores e programação-máquina. Arquitecturas: de von Neuman, paralela e distribuída. Processador: registos, obtenção da codificação de instrução, descodificação, unidade aritmética e lógica. Ciclo de execução, relógio, paralelização de ciclos (“pipelining”).

Departamento de Ciência de Computadores

Faculdade de Ciências

Universidade do Porto

Miguel Filgueiras, Sumários de “Introdução aos Computadores”

TP, semana de 25/10 Acesso e gestão de memória. Endereços: acesso directo, indirecto e indexado. Mecanismo de pilha: colocar (“push”) e retirar (“pop”) elementos, inversão de uma sequência de valores, operações aritméticas unárias e binárias, notação sufixa para expressões.

P, semana de 25/10 e dia 2/11 Construção de autómatos finitos simples.

27/10 Processadores: registos, registo de sequência, registo de condições. Codificação de instruções: tamanhos fixos ou variáveis, número e tipos de operandos. Tipos de instruções-máquina: mover (copiar) informação, aritmética, operações lógicas e sobre bits, deslocamentos e rotações, transferências de controle, instruções complexas. Variantes de instruções de acordo com tamanhos de operandos, alinhamento. Modos de endereçamento: literal, directo, indirecto, indexado, pré- e post-incremento ou decremento.

Memórias: volatilidade, tipos (RAM, ROM, EPROM, EEPROM, *flash*, memória central, memória para o sistema de arranque (BIOS), memórias auxiliares. Ordenação de informação guardada em mais de uma posição (“endianness”): arquitecturas “little-endian” e “big-endian”.

3/11 Linguagens de programação. Formas de programação primitivas: alteração de circuitos, uso de interruptores. Assembladores: macros. Linguagens de alto nível e de baixo nível. Paradigmas de programação: linguagens imperativas, linguagens funcionais, linguagens lógicas, linguagens orientadas para objectos; linguagens de comandos (“scripting”). Exemplos. Sintaxe de linguagens de programação: formas de representação, padrões, gramáticas, e diagramas sintácticos. Linguagens e gramáticas formais, classes de gramáticas.

TP, 3/11 e 8/11 Programas em linguagem-máquina mínima.

P, 4/11 a 9/11 Construção e teste de programas em Python ou C que implementam autómatos finitos simples.

8/11 Linguagens e gramáticas formais (cont.), classes de gramáticas, gramática independente de contexto, derivação, árvore de derivação, abreviaturas. BNF (Backus-Naur Form). Semântica: semântica operacional, semântica denotacional e semântica axiomática. Implementação de linguagens de programação: compiladores e interpretadores; comparação. Carregadores, código absoluto e código recolocável, compilação separada. Fases de compilação: análise léxica, análise sintáctica, geração de código intermédio, optimização, geração de código final. Inicialização de um compilador (“bootstrapping”).

10/11 Interpretadores, compilação para código binário não nativo, compilação no momento (“just in time”). Bibliotecas (“libraries”). Detectores de erro (“debuggers”), verificadores de correcção de programas, verificadores de tipos.

Exemplos: utilização do `gdb` na detecção de erros de programas em C; o interpretador de Tcl/Tk.

TP, 10/11 e 15/11 Gramáticas independentes de contexto: derivações, árvore de derivação, árvore sintáctica abstracta, ambiguidade, precedência de operadores.

P, 11/11 a 16/11 Programação em linguagem-máquina mínima.

Departamento de Ciência de Computadores
 Faculdade de Ciências
 Universidade do Porto

Miguel Filgueiras, Sumários de “Introdução aos Computadores”

15/11 Programação com linguagens de comandos e programação da casca (“shell”) em Unix. Ficheiros de comandos (“scripts”). Variáveis, argumentos de comandos, número de argumentos, lista de argumentos, comando **shift**. Substituição: escape (“\”), aspas, acentos agudos, acentos graves. Resultado da execução de um comando, comando **exit**. Comandos de programação: **if**, **for**, **while**, **case**. Leitura da entrada standard: comando **read**. Exemplos.

17/11 Continuação da aula anterior. Redireccionamento de entrada/saída: documentos locais (“here documents”, <<). “Pipelines” e listas de comandos (;, &&, ||). Comando **test**: testes sobre ficheiros, comparações de seqüências de caracteres, comparações de números, operadores lógicos. Comandos **expr**, **wc**, **sort**, **find**. Exemplos.

TP, 17/11 e 22/11 Expressões regulares. Concatenação, alternativa, e fecho de Kleene. Sintaxe (**man re_format**). Uso na linha de comando da “shell”, comando **grep**, exemplos.

P, 18 a 23/11 Programação da “shell”.

22/11 Sistemas de operação. Objectivos. Sistemas multi- e mono-utilizador, e multi- e mono-processo. Estrutura de sistemas de operação: camadas, núcleo (“kernel”). Interrupções por circuito e por software, rotinas de atendimento, Operações síncronas (execução sequencial, transferências de controle, chamada/regresso de subprogramas) e operações assíncronas; bloqueio. Processos, administração de memória e de ficheiros, chamadas ao sistema (“system calls”). Aplicações de sistema e aplicações de utilizador.

P, 25/11 a 30/11 Programação da “shell” (cont.).

29/11 Processos. Tempo partilhado: estratégias de temporização, *quanta*, prioridades. Processos: estados, acções, criação em Unix/Linux (“fork”, “execve”). O comando **ps** no Unix/Linux. Acções: suspensão, bloqueio, activação, morte. Comandos Unix/Linux relacionados com processos: **ps**, **top**, **kill**. Execução à frente (em “foreground”) e atrás (em “background”); comandos **fg**, **bg**, **jobs**; caracteres de controle na “shell” (interrupção **C**, suspensão, **Z**). Comunicação entre processos: ficheiros, canos, “sockets” locais, sinais (interrupções por software), mensagens. Sincronização de processos: ficheiros/variáveis de bloqueio (“lock”), semáforos, secções críticas de código.

TP, 29/11 e 15/12 Segurança e privacidade. Sistemas criptográficos de chave simples e dupla.

P, 2/12 a 7/12 Programação da “shell” (cont.).

6/12 Organização e administração de memória. Partições fixas e variáveis. Páginas: memória virtual e acesso a disco. Segmentos: permissões de acesso. Memória virtual: espaço de endereçamento virtual, mapeamento de endereços, unidade de gestão de memória do processador, troca (“swapping”), partições/ficheiros de “swap”, algoritmos de troca. Memória-cache.

Entrada/Saída. Acesso a periféricos através do sistema de ficheiros: ficheiros especiais, programas de controle de periféricos (“driver”). transmissão por caracter e por bloco. Operações sobre periféricos: inicialização, abertura, leitura, escrita e fecho. Interrupções.

P, 9/12 a 13/12 Resolução de problemas de exames anteriores.

Departamento de Ciência de Computadores

Faculdade de Ciências

Universidade do Porto

Miguel Filgueiras, Sumários de “Introdução aos Computadores”

13/12 Entrada/Saída (cont.). Armazenamento de informação: duplicação (“mirroring”; recuperação, disponibilidade), segmentação de ficheiros (RAID, distribuição de dados “data striping”; paralelismo), e redundância de informação (correção de erros). Independência relativamente ao equipamento. Sistemas de ficheiros: árvore de ficheiros, tipos de ficheiros, ligações simbólicas e duras, caminhos absolutos e relativos, volumes e montagem de sistemas de ficheiros. Transmissão em série e em paralelo. Acesso mutuamente exclusivo a periféricos: “spooling”.

15/12 Redes de computadores. Equipamento: linhas de transmissão de dados, conversões. Protocolos: codificação de informação, endereçamento de máquinas, transmissão por pacotes. Internet: endereços e nomes de máquinas; camadas física, de ligação de dados (“data link”), de rede, de transporte e de aplicações; referência a alguns protocolos.