

Programma degli argomenti svolti

anno scolastico: 2015/2016
classe: 3F (ITT)
materia: Tecnologia e Progettazione di Sistemi Informatici e di Telecomunicazioni
docenti: Prof. Lorenzo Donati, Prof. Mario D'Antoni

Nota: il presente programma non riflette pedissequamente la distribuzione temporale degli argomenti durante l'anno scolastico, bensì è organizzato secondo blocchi tematici omogenei. In grassetto sono indicati gli argomenti indispensabili per il raggiungimento degli obiettivi didattici minimi.

□ Sistemi di elaborazione digitali

○ Rappresentazione dell'informazione in un sistema informatico.

- ◆ Sistemi di numerazione posizionale. Sistemi decimale, binario, ed esadecimale. Conversioni tra basi di numerazione. Conversione diretta tra binario ed esadecimale.
- ◆ Rappresentazione macchina dei numeri naturali: codifica binaria naturale, codifica BCD.
- ◆ Rappresentazione macchina dei numeri interi: codifica in complemento a due, codifica in modulo e segno, codifica in traslazione.
- ◆ Rappresentazione macchina dei numeri reali: rappresentazione in virgola mobile ed errori di rappresentazione. Codifica IEEE754 in singola e doppia precisione.
- ◆ Cenni alla rappresentazione dei caratteri: codifica ASCII e codifica Unicode UTF-8.

○ Algebra di Boole

- ◆ Operazioni elementari AND, OR e NOT. Espressioni logiche. Espressioni logiche equivalenti ed identità logiche. Determinazione della tabella di verità di un'espressione logica. Cenni al concetto di funzione logica.
- ◆ Proprietà dell'algebra di Boole: associativa, commutativa, distributiva, doppia negazione, dei complementi, di idempotenza, di neutralità, di annullamento, di DeMorgan, di assorbimento, di assorbimento del complemento.
- ◆ Principio di dualità.

○ Reti logiche e struttura del calcolatore elettronico

- ◆ Porte logiche: AND, OR, NOT, buffer, NAND, NOR, XOR, XNOR. Circuiti logici.
- ◆ Determinazione della legge di commutazione di una rete combinatoria a partire dal suo schema logico e viceversa.
- ◆ Circuiti logici equivalenti. Porte a più ingressi. Universalità delle porte NAND e NOR.
- ◆ Circuito semisommatore (half-adder) e sommatore completo (full-adder).
- ◆ Multiplexer, demultiplexer e decoder binari.
- ◆ Reti riconoscitrici di stati logici. Riconoscimento a 1 e riconoscimento a 0. Mintermini e maxtermini. Forma canonica SP e forma canonica PS.

- ◆ Diagrammi di temporizzazione. Ritardo di propagazione nelle reti logiche.
- ◆ Azzardi (alee) nelle reti logiche. Circuiti riconoscitori dei fronti.
- ◆ Reti sequenziali sincrone ed asincrone: latch SR, latch D, flip-flop D, flip-flop T.
- ◆ Pilotaggio delle reti sequenziali: tempo di setup e tempo di hold.
- ◆ Uso di decoder binari per la decodifica degli indirizzi di memoria di un sistema a microprocessore (indirizzamento dei banchi di memoria).
- ◆ Registri di memoria con latch e flip-flop. Registro a scorrimento costruito con FF-D e sue applicazioni. Conversione seriale/parallelo mediante registro a scorrimento.
- ◆ Divisori di frequenza digitali mediante FF-T e loro applicazioni.
- ◆ Contatori asincroni (ripple counters) e sincroni.
- ◆ Cenni agli automi a stati finiti. Modello di Moore e modello di Mealy.

□ Linguaggi Informatici

○ Classificazione dei linguaggi

- ◆ Linguaggi dichiarativi – caratteristiche ed esempi: linguaggi di markup, linguaggi funzionali e linguaggi logici. Cenni storici ed applicativi sui seguenti linguaggi: HTML, LISP, Scheme, PROLOG, SQL, linguaggi degli spreadsheet, TeX/LaTeX, XML, PHP.
- ◆ Linguaggi imperativi – caratteristiche ed esempi: linguaggi di Von Neumann e linguaggi ad oggetti. Cenni storici ed applicativi sui seguenti linguaggi: Fortran, Basic, C, C++, Java, Forth, Cobol, Pascal, Perl, Javascript, Lua, Python.
- ◆ Concetto di linguaggio dominio-specifico (DSL – domain specific language) e linguaggio per la descrizione dei dati (DDL – data description language).

○ Caratteristiche ed implementazione dei linguaggi

- ◆ Esecuzione del codice: approccio compilativo, approccio interpretativo, approcci ibridi.
- ◆ Compilatori ed interpreti.
- ◆ Macchine virtuali ed interpreti di bytecode (Java, Javascript, Python, Lua). Compilazione JIT.

- ◆ Meccanismo di creazione del codice eseguibile a partire dal codice sorgente in C/C++ (usando il compilatore GCC come modello). Preprocessing; compilazione; linking statico. File oggetto. Cenni al linking dinamico.

□ **Fondamenti sui sistemi operativi**

○ **Concetti generali**

- ◆ Concetto di struttura dati con disciplina di gestione LIFO: gli stack in astratto e lo stack hardware gestito dalla CPU. Problemi di gestione di uno stack (overflow, underflow).
- ◆ Cenni all'utilizzo dello stack hardware per l'implementazione del meccanismo della ricorsione e delle variabili locali in C/C++.
- ◆ Concetto di condivisione e gestione delle risorse del sistema. Sistemi operativi monoutente, multiutente, uniprocesso, multiprocesso.
- ◆ Overhead di sistema.
- ◆ Protocolli di comunicazione.
- ◆ Dispositivi fisici e logici. Driver di dispositivo.
- ◆ Tecniche di caching e di buffering.
- ◆ Serializzazione, spooling e code di stampa.
- ◆ Chiamate di sistema.
- ◆ Filesystem.
- ◆ Mappa di memoria logica e mappatura con la memoria fisica.
- ◆ Meccanismo delle interruzioni: vettore delle interruzioni, subroutine di gestione delle interruzioni, mascheramento delle interruzioni.
- ◆ I/O sincrono ed asincrono. Controllore DMA.
- ◆ Sistemi multiprocesso e time sharing. Parallelismo virtuale e reale. Commutazione di contesto e PCB (process control block). Stati di un processo (nuovo, pronto, attivo, in attesa, defunto). Scheduling dei processi.

- ◆ **Processi genitori e processi figli.** Funzioni di sistema *fork* e *wait*.
- ◆ Servizi e processi eseguiti in background.
- ◆ **Comunicazione interprocesso (IPC):** comunicazione mediante passaggio di messaggi e mediante memoria condivisa.
- ◆ **Sicurezza dei sistemi e nella gestione dei processi:** autenticazione degli utenti, livelli di privilegio degli utenti, livello di privilegio nell'esecuzione di codice, modalità privilegiata della CPU, codice di sistema e codice utente, istruzioni macchina privilegiate. Contabilità delle risorse e gestione delle quote.
- ◆ **Threads e multiprogrammazione:** vantaggi e svantaggi.
- ◆ Cenni alle problematiche di sincronizzazione di processi e threads. Cenni alle sezioni critiche.

□ **Strumenti di sviluppo e gestione dei sistemi**

○ **Shell di Windows**

- ◆ Introduzione all'uso dell'interprete dei comandi di Windows.

○ **Task Manager**

- ◆ Dimostrazione dell'uso di ProcessExplorer (task manager freeware avanzato per i sistemi Windows) per esaminare i processi in esecuzione su di una macchina.

○ **Debugger**

- ◆ Uso del debugger per analizzare il codice macchina di un programma durante l'esecuzione.
- ◆ Concetto di codice disassemblato e cenni alla sintassi del linguaggio Assembly per i processori x86 (sintassi INTEL).