Informatica prima lezione

Esempi delle istruzioni (slide 30)

I codici con j all'inizio stanno per salto per velocizzare i processi
Push e pop servono per mettere o leggere in memoria informazioni
Call serve per saltare al sottoprogramma
Halt serve per terminare
Ogni istruzione ha 2 parti uno il codice operativo e gli altri sono gli operandi
Ci sono istruzioni effettive ovvero somme e sottrazioni e altre sono istruzioni di controllo
Le istruzioni a livello di codice macchina solo solo messi in 0 e 1
Abbiamo un certo numero di bit per la quantità di istruzioni diverse mentre gli altri bit vengono utilizzati per gli operandi
Questo linguaggio si chiama assembly ed è l'unico che il sistema di calcolo può comprendere

Piramide della gerarchia di memoria
Tempi di accesso più corti sono a piccola capacita e più costosi e stanno in alto
Cpu è In cima alla piramide

Seconda lezione

Sistema operativo
Insieme di programmi che accedono alla macchina fisica e hanno il compito di utilizzare il componente fisico rendendolo disponibile agli stati software sovrastanti
Nel fare cio rende disponibili le risorse con una visione semplificata (riduce la complessità delle risorse agli stati superiori) ed estesa
Agli strati superiori è più facile leggere un informazione che è nella memoria di massa mai non dobbiamo sapere se è in un disco magnetico o qualsiasi altra forma di archiviazione quindi maschero la complessità di questi dettagli .

Esteso per quanto riguarda le potenzialità
Dal punto di vista delle applicazioni pensano di aver a disposizione di una quantità illimitata di spazio. L'applicazione si comporta come se avesse tutte le risorse disponibili.

Ogni elemento di un sistema operativo ha in gestione una risorsa per i programmi
Utilizzerà meccanismi e politiche per raggiungere determinate prestazioni.

Risorsa del sistema operativo è la cpu ovvero il gestore dei processi (cernell) strato più vicino alla macchina fisica.

Cos'è un processo? Un programma in esecuzionione ovvero dati istruzioni e stato dell'esecuzione.
Dato un programma puo dar vita a più processi.
All'interno del sistema operativo il kernel gestisce il processore e manda programmi superiori ad uno.
Condivisione e suddivisione della gestione del tempo
La cpu è sempre attiva il kernel gestisce il tempo della cpu per le diverse applicazioni.
In questo modello quando creo un nuovo modello esso è pronto ma non utilizza ancora la cpu poiché lavora su un processo alla volta. Una volta che la cpu viene affidata all'applicazione è in running.
Il processo ad un certo punto finisce e dunque termina
Se un processo è in attesa di un dato esterno è in stato di wait

La politica quindi utilizzata è quella di consentire a tutti un quanto di tempo per cui ogni programma entra in stato di running per poi tornare in stato di ready (round deboding)
Quindi sembra che ogni applicazione lavori in parallelo.
Se il quanto di tempo è gestito male si avrà una gestione dei tempi fatta male e le applicazioni sembreranno lente.
Come pianifico il tempo è lo skedding

Gestore della memoria di lavoro:
Semplifica: non necessario sapere dove sono fisicamente caricati dati e istruzioni ed non è necessario preoccuparsi di proteggere l'accesso ai propri dati
Estende: è possibile eseguire un qualsiasi programma indipendentemente dalla quantità di memoria necessaria e dal numero di programmi in esecuzione
È come se avesse a disposizione tutta la memoria necessaria.

Quando il numero di indirizzi logici è maggiore del numero di indirizzi fisici non c'è sufficiente memoria di lavoro per tutte le pagine
Una parte della memoria secondaria viene utilizzata per salvare le pagine dei processi in stato di pronto o sospeso.
Lo spostamento di pagine tra memoria di lavprp e memoria di massa è lo swapping

La visione offerta dal gestore della memoria di lavoro è:
Memoria di capacita infinita
Veloce e non volatile
Ma effettivamente la memoria è insufficiente per tutti i processi attivi e la memoria è insufficiente per un solo processo enorme
Si sitroduce dunque la memoria cache perché più veloce della memoria principale, essa sfrutta la velocità della memoria cache e le dimensioni della memoria principale
Si usa il meccanismo della memoria virtuale per tenere aperte delle informazioni dei programmi in esecuzione stanno su memoria si massa e sfrutta la velocità della memoria principale e le dimensioni della memoria secondaria

Terza lezione

Rappresentazione dell'informazione
Trovare un modo opportuno per rappresentare all'interni di un sistema di calcolo le informazioni (dati e programmi)
In modo efficace rispetto a: realtà fisica del sistema e la loro manipolazione

Gli elementi di rappresentazioni sono l'alfabeto ovvero ii simboli che possono essere utilizzati e i codici ovvero le regole che definiscono quali sequenze di simboli sono ammissibili o no.
Cio che è ammissibile è definito parola di codice
Quando guardo una sequenza di simboli riesco a dire il suo significato e l'appartenenza.

Il codice definisci quali sequenze sono ammissibili e la loro relazioni tra gli elementi in una sequenza ammissibile

Le configurazioni ammissibili hanno le stesse dimensioni e la dimensione dipende da

Abbiamo l'alfabeto dei simboli s = ()
Cardinali dell'alfabeto
Numero di elementi da rappresentare n
Dimensioni delle configurazioni K
K=(logsn)

Il numero di configurazioni diverse di lunghezza k: n = s^k

I componenti elettronici sono costituiti da condensatori (carichi scarica), linea di tensione (alta bassa)

La mappatura diretta con un sistema costituito da due simboli ovvero il sistema binario (0,1)
Per qualsiasi cosa come valori numerici simboli immagini suoni ecc…

Il codice binario
Si avvale di un sistema fatto da due simboli (0,1)
Esse sono chiamate cifre della codifica non che binary digit che abbreviato è bit
Byte: 8 bit
Kilobite: 2^10
Megabite: 2^20
Gigabite: 2^30
Terabyte: 2^40

Dati:
L'insieme degli elementi da rappresentare: i semi delle carte da gioco - l'insieme delle configurazioni ammissibili il codice definisce l'associazione biunivoca

Scelta del codice: binario
S: (0,1)
/S/ = 2

Logs N quindi log2 4 = 2

L'insieme delle configurazioni ammissibili 00, 01, 10, 11
Il codice definisce l'associazione biunivoca

Se abbiamo i giorni della settimana: lu ma me gio ve sa do
L'insieme delle configurazioni ammissibili: 000, 001, 010, 011, 111, 110, 101

Aver scelto un alfabeto non è sufficiente ma è necessario scegliere anche un codice
Se conosciamo l'alfabeto utilizzato non siamo ancora in grado di comprendere quale sia l'informazione rappresentata

Informazione non numerica
Obbiettivo è quello di definire una codifica per ogni possibile carattere rappresentabile
Inizialmente rea usato l'alfabeto base
Per poi passare ai caratteri numerici da 1 a 9
simboli. Di interpunzione . , : …
E poi i caratteri speciali come l'scapo nuova riga e spazio
In tutto erano 120 elementi
Lunghezza codificata logs n = log2 120 = 7
Il primo codice era il codice ASCII base fatto su 7 bit
Per poi definire iol codice ASCII esteso con base 8 bit
Esso aveva interno a lui il codice ASCII base per le prime 128 configurazioni più i caratteri nazionali (à è é), simboli e cornici. Questi sono chiamati caratteri semigrafici.

Dallo 0 al 32 sono simboli non stampabili

Ad ogni singolo carattere corrisponde un bYte
I caratteri adiacenti alfabeticamente hanno codifiche adiacenti e i caratteri maiuscoli e minuscoli non sono la stessa cosa (le maiuscole vengono prima delle minuscole)

Del codice ASCII bisogna ricordare:
I caratteri numerici vengono prima dei caratteri alfabetici e le maiuscole prima delle minuscole

Codice UNICODE
Esso è standard e internazionale che permette di rappresentare tutti i caratteri attraverso un codice a 16 bit che raggruppa tutti gli alfabeti esistenti, anche retrocompatibile con il codice ASCII
Rispetto al codice ASCII le dimensioni dei file sono raddoppiati

Numeri interi
Il numero naturale è un oggetto matematico che può essere rappresentato come una sequenza di simboli a partire da un alfabeto:
Base 10: (0,9)
Base 5: (0,4)
Base 2: (0,1)

Rappresentazione additiva con i valori romani
Rappresentazione posizionale con simboli che vanno da 0 a 9. Il ruolo posizione: reso rispetto alla base b

Abbiamo la base ovvero l'insieme dei simboli

Sistema binario
Per quanto riguarda l'aritmetica è sempre consentita, la sottrazione solo se il sottraendo è minore del minuendo
La divisione solo se esiste un valore quoziente tale che moltiplicato per il divisore da il dividendo

Conversione da diverse basi
Abbiamo due strade
Il metodo della somma dei prodotti per le basi elevate alla potenza in base alla posizione
Il metodo del resto delle divisioni rispetto alla base

Notazione in modulo e segno
Per rappresentare i numeri interi relativi
La notazione più semplice che si adotta nel sistema decimale è quella modulo e segno
Con segno ovvero un simbolo per siscriminaew tra valori positivi e negativi
Doppia codifica del valore 0: ovvero +0 e -0

Sulla base 2 per il segno si usa lo 0 per il + e l'1 per il -
Positivo 5 = 0101

Abbiamo 4 bit — 2^4 = 16 configurazioni
(-7,+7)
(1111, 1111)

Dato un valore intero relativo x, la sua rappresentazione in base b
Un elemento cifra per rappresentare il modulo

Data la codifica in base 2 su 6 cifre quelle intervallo di valori interi relativi consente di rappresentare una cifra per positivo/negativo
Valori in modulo rappresentabili 2^(6-1) - 1

Per non confondersi tra numero in base 2 e numero in base due ma con segno viene affiancato alla base di due una sigla di MS

Analisi
Conversione da base 10 modulo e segno