

Il Sistema Operativo

Fabio Massimo Zanzotto

Ricapitoliamo puntate precedenti

- Cosa abbiamo a disposizione:
 - Concetto di algoritmo (con eventuale parametrizzazione) (ALGO)
 - Rappresentazione dell'informazione (RAP)
 - Un risolutore generale di problemi
($RGP = RAP + ALGO + ALGO_{VITALE}$)
- Problema:
 - Come usiamo questo Risolutore Generale di Problemi?

Cosa abbiamo costruito fino ad adesso?

Linguaggio L
SPOSTA <OP_A> <OP_B>
SALTA <OP_A>, SALTA_SE<_> <OP_A>, ...
SOMMA <OP_A> <OP_B>
COMPARA <OP_A> <OP_B>

Programma_L

Macchina che parla il
Linguaggio L

University of Naples "Ter Vergata"

Idee Fondamentali per costruire la macchina di Von Neumann

- Algoritmo Vitale:** Si può pensare ad un algoritmo "vitale" che abbia come scopo quello di eseguire algoritmi
- Dati e algoritmi:** Un algoritmo scritto con una certa codifica può essere "dato" di un altro algoritmo

F.M. Zucchetto, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

University of Naples "Ter Vergata"

Conseguenze delle idee fondamentali

"Confusione" tra Dati&Programmi

Programma_L

traduttore

Programma_L

Macchina che parla il Linguaggio L

F.M. Zucchetto, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

University of Naples "Ter Vergata"

Conseguenze delle idee fondamentali

"Confusione" tra Dati&Programmi

Macchine Astratte

Macchine Astratte per Servizi di Base

Sistema Operativo

Macchine Astratte per le Attività più diverse

Gestione Memoria

Gestione CPU

Gestione IN-OUT

Ritorniamo alle macchine specifiche!!!

F.M. Zucchetto, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

2

University of Naples "San Vito"



Come procediamo

- Definizione di Sistema Operativo
 - Chi serve: Analisi utenti del SO
 - Cosa controlla
 - Come lo controlla
- Macchina astratta
- Moltiplicazione risorse: macchine virtuali
- Moltiplicazione risorse: come otteniamo il miracolo?
- Gestione risorse
 - Memoria Centrale
 - Memoria di Massa
 - Dispositivi di I/O

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

Il Sistema Operativo

Un insieme di macchine astratte

University of Naples "San Vito"



Sistemi Operativi

Cos'è un sistema operativo ?

- Un sistema operativo e':
 - La astrazione logica del livello macchina hardware e microprogrammata
 - Un insieme di programmi che rende "utilizzabile" un sistema di elaborazione.
 - Il gestore delle risorse di un sistema di elaborazione
 - L'interfaccia utente di un qualsiasi elaboratore

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia





Chi serve: Utenti di un SO

- Nella classe degli utenti di un sistema operativo troviamo:
 - Operatori dei sistemi di elaborazione
 - Programmatori di Sistema
 - Programmatori di Applicazioni
 - Personale Amministrativo
 - Utenti
 - Programmi
 - Sistemi/Dispositivi Hardware

F.M. Zuccone, R. Basili

Abilità Informatiche

Facoltà di Lettere e Filosofia





Cosa controlla: Gestione Risorse

- Un Sistema Operativo gestisce in genere un sistema le cui risorse essenziali sono:
 - Uno o più Processori
 - Memoria
 - Dispositivi di Memorizzazione di Massa
 - Dispositivi di I/O
 - Dati

F.M. Zuccone, R. Basili

Abilità Informatiche

Facoltà di Lettere e Filosofia





Come controlla: Gestione Risorse (2)

- Garantire l'utilizzo del sistema significa:
 - Fornire una astrazione adeguata dei singoli dispositivi agli utenti dei livelli macchina superiori
 - Controllare l'accesso condiviso a tutte le risorse
 - Garantire tolleranza agli errori

F.M. Zuccone, R. Basili

Abilità Informatiche

Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Come controlla: Gestione Risorse (3)

- In quanto gestore delle risorse un S.O. svolge le seguenti funzioni:
 - Fornisce una interfaccia all'utente (comandi/grafici/icone)
 - Regola la condivisione di risorse tra gli utenti
 - Effettua la allocazione delle risorse agli utenti che le richiedono
 - Garantisce una astrazione delle funzioni di I/O
 - Gestisce Errori: Rilevamento, Reazione, Eliminazione, Ripristino

F.M. Zuccone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

Macchina Astratta

University of Naples "San Vito"

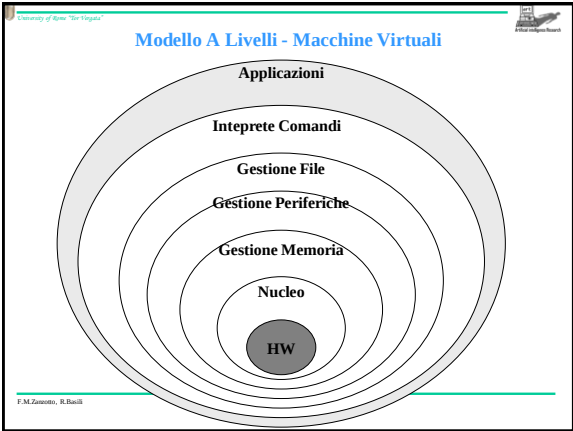


Sistema Operativo: macchina astratta

Sistema Gestione Biblioteca	Giochi	Sistema Informativo Bancario
Compilatori	Editor	Interprete Comandi
Sistema Operativo		
Linguaggi Macchina		
Microprogrammazione		
Dispositivi Fisici		

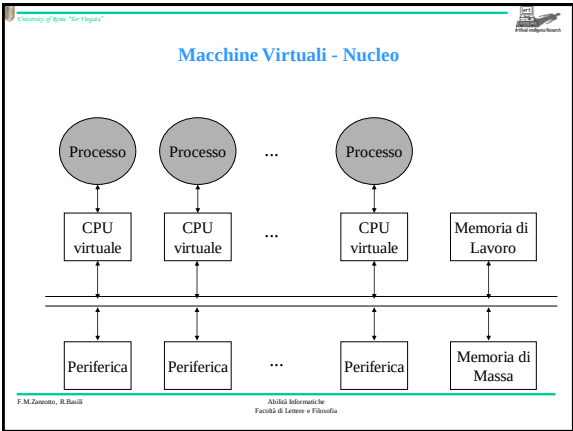
F.M. Zuccone, R. Basili

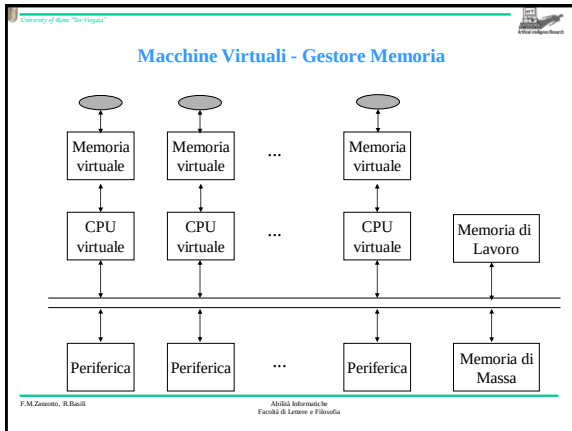
Abilità Informatiche
Facoltà di Lettere e Filosofia

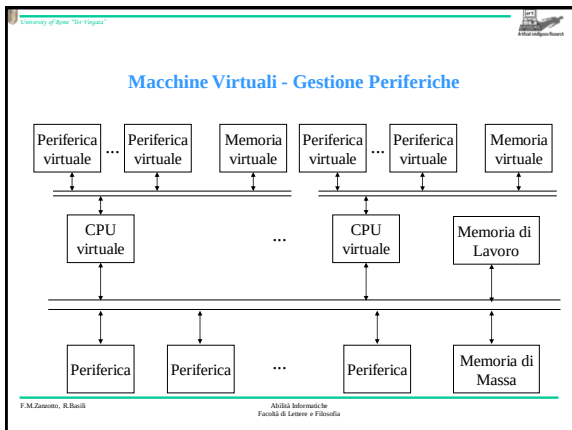


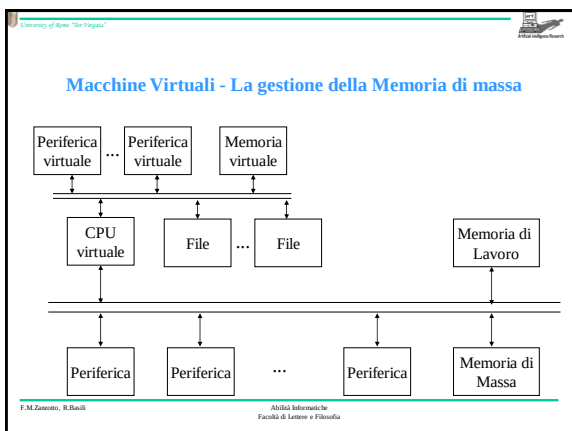
Macchine Virtuali

Moltiplicazione delle risorse









University of Naples "San Vito"

PIRELLA GÖTTSCHE LOWE

Interfaccia Utente

Ogni Sistema Operativo ammette una componente che viene generalmente denominata interprete dei comandi o *shell*. La *shell* garantisce una serie di primitive ad alto livello (i comandi) che supportano l'interazione tra l'utente e la macchina, permettendo cioe':

- l'attivazione dei comandi;
- l'accesso alla memoria di massa;
- l'accesso ai dispositivi di I/O;
- alcune funzioni ausiliarie (per es. *pipelining*);

F.M.Zanotto, R. BasiliAbilità InformaticheFacoltà di Lettere e Filosofia

University of Naples "San Vito"

PIRELLA GÖTTSCHE LOWE

Interfaccia Utente: Esempi

- Esempi tipici:
 - **Command.COM** nel Ms-DOS

```
c:\miadir> type pippo.txt
c:\miadir> dir /p
c:\miadir> mkdir documenti
```
 - **Cshell** in Unix

```
/usr/local/user> cat pippo.txt
/usr/local/user> ls -al | more
/usr/local/user> md documenti
```

F.M.Zanotto, R. BasiliAbilità InformaticheFacoltà di Lettere e Filosofia

University of Naples "San Vito"

PIRELLA GÖTTSCHE LOWE

Interfaccia Utente

- Le Interfacce grafiche: Windows

F.M.Zanotto, R. BasiliAbilità InformaticheFacoltà di Lettere e Filosofia

Processi e Moltiplicazione Macchine Virtuali

Come otteniamo la moltiplicazione delle risorse?

Moltiplicazioni delle risorse: **il processore**

Come è possibile far “credere” ad ogni programma/processo di avere un processore tutto per sé?

Oppure, come possiamo servire più processi in un tempo finito e far finta che siano eseguiti contemporaneamente?

Osserviamo la realtà
Come facciamo a “dare” quattro esami in un appello?

Processi e Moltiplicazione Macchine Virtuali

Obiettivo:

“vogliamo ottenere che con un unico processore possano essere eseguiti più programmi”

Mezzo:

Multiprogrammazione

Concetto Fondamentale:

Processo

Come otteniamo la moltiplicazione delle macchine:

dividendo il tempo!





Problema

Il sistema operativo deve decidere:

avendo una sola CPU come è possibile soddisfare numerosi processi

ovvero:

- decidere se il processo che sta usando la CPU può continuare, ed eventualmente sospenderlo
- decidere il successivo processo a cui concedere l'uso della CPU
- gestire una coda di processi attivi

F.M. Zuccone, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Domande

- **Il processo: questo sconosciuto**
Come deve esser fatto un processo se vogliamo interrompere la sua esecuzione?
- **La gestione della risorsa CPU-Scheduling**
Come possiamo gestire la coda dei processi che combattono per la CPU?

F.M. Zuccone, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Il processo: questo sconosciuto

- Un processo è un **programma in esecuzione**, descritto cioè da un suo stato:
 - istruzione successiva da eseguire
 - stato dei propri dati
 - dallo stato di eventuali processi ad esso correlati (ad es. figli)

F.M. Zuccone, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Il processo: questo sconosciuto

- Un processo sequenziale puo' essere:
 - ATTIVO**: mentre le istruzioni del suo programma vengono eseguite
 - IN ATTESA**: mentre attende che alcuni eventi abbiano luogo (per esempio, che un altro processo rilasci la stampante)
 - PRONTO**: quando attende di essere assegnato ad un processore, cioe' attivato.

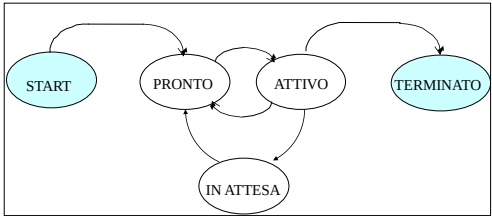
F.M.Zanetto, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Il processo: questo sconosciuto



```
graph LR; START((START)) --> PRONTO((PRONTO)); PRONTO --> ATTIVO((ATTIVO)); ATTIVO --> IN_ATTESA((IN ATTESA)); IN_ATTESA --> PRONTO; PRONTO --> START; ATTIVO --> TERMINATO((TERMINATO));
```

F.M.Zanetto, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Il processo: questo sconosciuto

Ogni processo e' rappresentato dal suo **Blocco di Controllo dei Processi (Process Control Block)** che contiene informazione riguardo a

- Stato del processo
- Program Counter
- Registri della CPU
- Informazioni per la schedulazione (priorita', puntatore al successivo, ...)
- Informazioni per la gestione della memoria del processo
- Stato dei canali di I/O, dei file aperti, delle richieste

F.M.Zanetto, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

INFORMATICA

Il processo: questo sconosciuto

- Il PCB di un processo

Puntatore	Stato del Processo
PID Processo	
Program Counter	
Registri	
Limiti di Memoria	
File Aperti	
* * *	

F.M.Zanotto, R.BaciliAbilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

INFORMATICA

Avanzamento di un processo

P0 **Sistema Operativo** **P1**

Salva lo stato in PCB0
...
Ripristina lo stato da PCB1

Salva lo stato in PCB1
...
Ripristina lo stato da PCB0

tempo

F.M.Zanotto, R.BaciliAbilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

INFORMATICA

Domande

- Il processo: questo sconosciuto**
Come deve esser fatto un processo se vogliamo interrompere la sua esecuzione?
- La gestione della risorsa CPU-Scheduling**
Come possiamo gestire la coda dei processi che combattono per la CPU?

F.M.Zanotto, R.BaciliAbilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"



La gestione della risorsa CPU-Scheduling

Nei sistemi uniprocessore anche se molti processi sono attivi solo UNO puo' ottenere il controllo della CPU. E' necessario definire metodi per decidere:

- quale processo
- per quanto tempo

merita o puo' ottenere il controllo della CPU.

Tale compito e' detto SCHEDULAZIONE (o scheduling).

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"



La gestione della risorsa CPU-Scheduling

L'algoritmo

1. Esegui un processo P fino a quando non puo' evolvere oltre (ad es. richiede una risorsa non disponibile)
2. Allora togli al processo P il controllo della CPU
3. Scegli tra i processi PRONTI il nuovo processo Q a cui affidare il controllo della CPU
4. Torna in 1 con P uguale a Q.

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"



La gestione della risorsa CPU-Scheduling

Processo P  a) **1 2 processi visti separatamente**

Processo Q 

b) **1 2 processi senza multiprogrammazione**

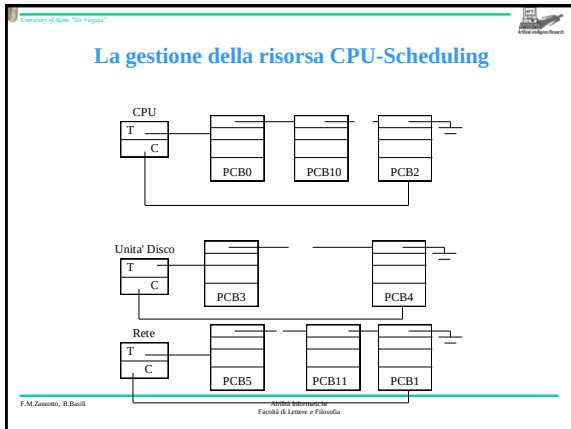
Processo P  Processo Q 

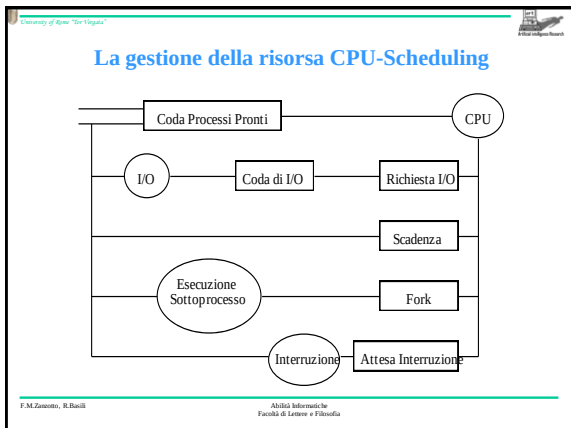
c) **1 2 processi in un ambiente a multiprogrammazione**

Processo P  Processo Q 

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia





University of Rome "Tor Vergata"

La gestione della risorsa CPU-Scheduling

Lo *scheduling* puo' aver luogo in una delle quattro situazioni seguenti:

- Il passaggio di un processo da ATTIVO in PRONTO
- Il passaggio di un processo dallo stato di IN ATTESA ad uno stato di PRONTO (ad es. e' avvenuto un *interrupt*)
- Il passaggio di un processo dallo stato di ATTIVO ad uno stato di IN ATTESA (ad es. e' avvenuto un *interrupt*)
- Un processo termina

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Politiche di Scheduling

- Le tecniche algoritmiche per la gestione dello *scheduling* si possono raggruppare in alcune classi piu' importanti:
 - algoritmi FIFO
 - algoritmi Shortest-Job-First
 - algoritmi basati su priorita'
 - algoritmi *Round-Robin*

F.M.Zanetti, R.Bacchi

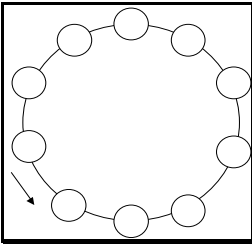
Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Politiche di Scheduling

- Round-Robin
 - coda circolare
 - quanto di tempo, τ
 - ad ogni processo non piu' di un quanto τ



F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Round-Robin

Processo	Tempo di CPU
P1	24
P2	3
P3	3

P1	P2	P3	P1	P1	P1	P1	P1
----	----	----	----	----	----	----	----

4 ms

3 ms

Tempo medio d'attesa = 5.6 ms

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Scheduling: problemi tipici

Starvation

– *Starvation* si riferisce alla situazione in cui un processo che non puo' avanzare, poiche' in attesa di una risorsa, attende indefinitamente senza mai terminare la propria computazione.

– Come sottolineato gli algoritmi di scheduling consentono di evitare tali situazioni (es. Round-Robin)

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Storia - Multiprogrammazione (Parallelismo)

• La disponibilità di hardware sempre piu' potente e costoso, e la variabilità dei compiti dei programmi condussero negli anni '70 all'introduzione del concetto di multiprogrammazione.

• In un sistema multiprogrammato, più programmi possono essere eseguiti contemporaneamente anche se il processore e' in grado di eseguire sequenzialmente le istruzioni di un solo programma per volta.

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Storia - Multiprogrammazione (Parallelismo)

• La residenza in memoria di piu' programmi attivi, che si alternano nel controllo della CPU, e' detta multiprogrammazione.

• La nozione di processo cattura adeguatamente il concetto di programma attivo.

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

Logo

Interazione tra Processi

- Modalita' di un processo
 - modo utente
 - modo supervisore
- Processi foreground
- Processi background

F.M. Zuccone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

Processi e Convivenza

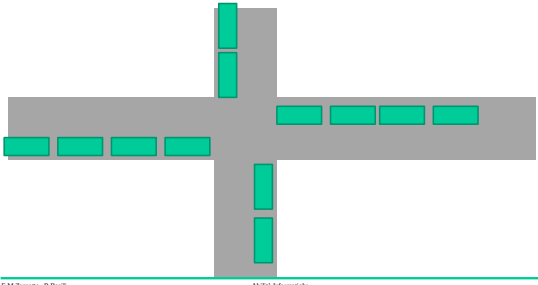
Problemi tipici

University of Naples "San Vito"

Logo

Deadlock

Osserviamo la vita reale!



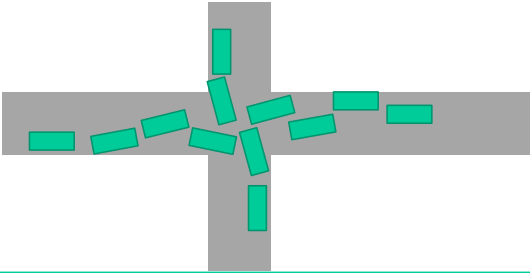
F.M. Zuccone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

Deadlock

Osserviamo la vita reale!



F.M. Zuccone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

Deadlock: modelliamo



F.M. Zuccone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"

Deadlock

- 5 filosofi passano la vita pensando e mangiando
- I filosofi condividono un tavolo rotondo con 5 posti.
- Un filosofo per mangiare deve usare due bacchette



F.M. Zuccone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

18

University of Rome "Tor Vergata"

Deadlock

- Esistono più processi che vogliono accedere a delle risorse
- Il deadlock avviene quando questi decidono di accedere alle stesse risorse

```

graph LR
    P1((P1)) -- ha --> R1[Risorsa1]
    P1 -- vuole --> R2[Risorsa2]
    P2((P2)) -- ha --> R2
    P2 -- vuole --> R1
  
```

F.M. Zucchetto, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"

Corse Critiche

Processo 1
 Leggi A
 $A = A + 100$
 Scrivi A

Memoria
 A
 50

Processo 2
 Leggi A
 $A = A - 100$
 Scrivi A

F.M. Zucchetto, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"

Corse Critiche

- Sono corse critiche quelle situazioni in cui il risultato della esecuzione delle istruzioni di due programmi che si contendono l'accesso ad una risorsa dipende dall'ordine di esecuzione delle situazioni dei due programmi.
- La sequenza delle istruzioni che contenendo accesso a risorse condivise possono generare corse critiche e' detto sezione critica.
- Tipiche sezioni critiche sono relative a problemi di mutua esclusione tra i processi.

F.M. Zucchetto, R. Basili

Abilità Informatiche
 Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Corse Critiche

REPEAT

Entrata

Sezione Critica

Uscita

Programma rimanente

UNTIL false

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Corse Critiche

- Soluzioni
 - Con *mutua esclusione* (meno efficienti)
 - Variabili di Lock
 - Senza *mutua esclusione*
 - Semafori
 - Monitor
 - Scambio di messaggi

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Accesso a risorse condivise

Problemi tipici:

- Deadlock: si ha un ingorgo
 - Soluzione: impedire ai processi di attendere all'infinito una risorsa
- Corse critiche: si hanno risultati sbagliati
 - Concetto di *mutua esclusione*

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

Gestione Memoria Centrale

Un insieme di macchine astratte

Moltiplicazione della memoria centrale

Obiettivo:

“vogliamo far in modo che la memoria centrale di una macchina possa ospitare più processi PRONTI di quanti in realtà ne può ospitare”

Concetto Fondamentale:

I processi sono in differenti stati (PRONTO, ATTIVO, IN ATTESA)

Come otteniamo la moltiplicazione della memoria centrale:

~~dividendo il tempo!~~

Usando oculatamente la memoria secondaria

Gestore Memoria

- Il sistema operativo garantisce ai programmi utente e supervisor
 - il caricamento di essi in memoria centrale
 - la ottimizzazione dell'uso della memoria, mantendendo in memoria solo la componente dei dati e del programma utile in una certa fase di elaborazione
- Il gestore della memoria garantisce quindi uno spazio di indirizzamento virtuale ai programmi piu' grande dello spazio fisico di memoria disponibile

University of Rome "Tor Vergata"

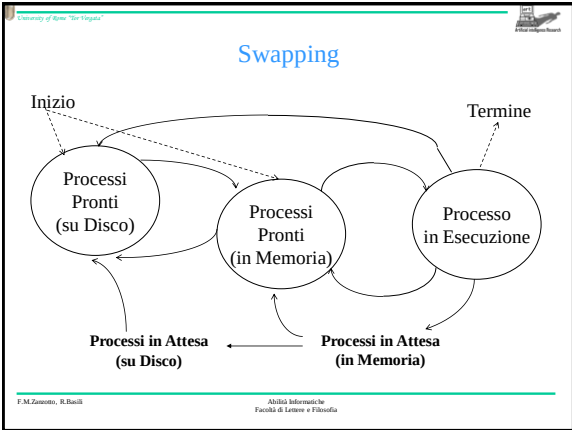


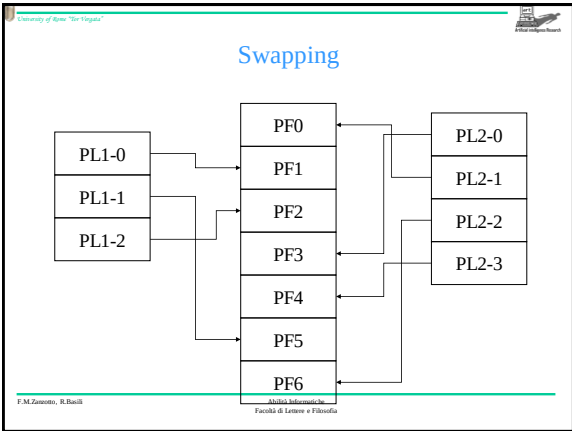
Gestione della Memoria: swapping

- Lo spostamento di processi dalla memoria al disco e viceversa e' necessaria in sistemi multiprogrammati E timesharing poiche' la memoria e' insufficiente per contenere il numero (impredicibile) dei processi utente. Questa' attivita' e' detta SWAPPING (scambio).
- Due tecniche che consentono un efficace gestione della memoria in presenza di swapping sono la gestione virtuale della memoria (o memoria virtuale) e la segmentazione.

F.M.Zanetti, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia



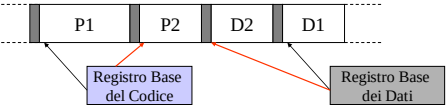


University of Applied Sciences "San Vito"



Gestione della memoria

- **Rilocabilità' del codice**



F.M.Zanotto, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Applied Sciences "San Vito"



Gestione della Memoria

- **Segmentazione**
 - I vari processi non hanno una dimensione fisica qualsiasi
 - Per evitare un uso ridondante della memoria essi vengono suddivisi in differenti componenti di dimensione fissata e di uso potenzialmente condiviso tra processi diversi (per es. librerie grafiche)
 - I componenti vengono detti segmenti:
 - codice
 - dati
 - pila
 - I segmenti vengono gestiti in modo paginato

F.M.Zanotto, R.Bacchi

Abilità Informatiche
Facoltà di Lettere e Filosofia

Gestione Memoria di Massa

University of Rome "Tor Vergata"

Gestione Memoria di Massa

- Le operazioni tipiche effettuate sulla memoria di massa sono:
 - Recupero di Informazione precedentemente memorizzata
 - Eliminazione della Informazione
 - Modifica/Aggiornamento dei dati
 - Copia o trasferimento (es. da Hard-disk a floppy-disk)
- Il gestore della memoria di massa che garantisce una visione logica delle suddette operazioni e' anche detto **File system**

F.M.Zanotto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"

Il File System

- Localizzazione delle Informazioni
 - Ogni informazione memorizzata e' organizzata in un File (archivio)
 - Nome
 - Estensione
 - I file sono raggruppati in insiemi e organizzati gerarchicamente
 - Directories
 - Struttura ad albero

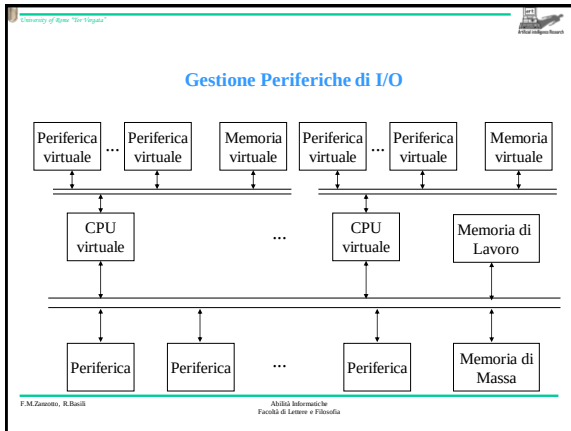
Informazioni generali e Filosofia

File

F.M.Zanotto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

Gestione Periferiche di I/O



University of Rome "Tor Vergata" 

Gestione Periferiche I/O

- I/O indipendente dall'hardware sottostante
 - driver fisici
 - specifici al tipo di dispositivo (HD vs. CD-ROM)
 - specifici al tipo di costruttore di un certo HW
 - driver logici
 - realizzano primitive di accesso (I/O) indipendenti dal tipo di dispositivo (files su CD-ROM e su HD)
 - sono da installare/configurare oppure **Plug&Play**
 - garantiscono la affidabilità' del processo (generalmente asincrono)

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata" 

Gestione Periferiche I/O

- Tipici dispositivi sono
 - video
 - stampante/i
 - tastiera
 - CD-ROM
 - Porte esterne
 - dispositivi dedicati (es. tastiere MIDI)
 - rete telefonica
 - audio

F.M. Zucchetto, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Gestione Periferiche di I/O

- Un esempio: la stampa
 - Una stampante non è condivisibile
 - La stampa viene *virtualizzata* in stampanti dedicate
 - I conflitti vengono gestiti mediante lo *spooling*
 - una richiesta di accesso (per la stampa) viene accodata
 - le stampe vengono effettuate mediante una politica FIFO fino allo svuotamento della coda (detta coda di stampa)
 - la attesa attiva in memoria permane finché una nuova richiesta viene accodata

F.M.Zanotto, R.Basili

Abitilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Gestione Periferiche di I/O

- Un esempio: uso della redirectione in Unix (1)

```
ELETTRA - Basili >>ls -al
-rw-r--r-- 1 basili users 100 Sep 14 15:35 .login
-rw-r--r-- 1 basili users 254 Sep 14 15:35 .cshrc
drwxrwxrwx 2 basili users 512 Sep 23 18:19 Japan#
drwxr-xr-x 2 basili users 512 Nov 29 11:54 Lyras
drwx----- 2 basili users 512 May 6 1999 Mail
-rw-r--r-- 1 basili users 0 Sep 14 15:35 dead.letter
-rw-rw-rw- 1 basili users 1698 Dec 6 18:27 sample.txt
-rw-rw-rw- 1 basili users 20088 Dec 6 18:26 sample.txt.mor
-rw-rw-rw- 1 basili users 3554 Dec 6 18:33 sample.txt.tok
ELETTRA - Basili >> cat sample.txt
```

F.M.Zanotto, R.Basili

Abitilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Gestione Periferiche di I/O

- Un esempio: uso della redirectione in Unix (2)

```
ELETTRA - Basili >>cat sample.txt
The Chief Engineer is personally responsible to supervise the bunkering
operations at all times and for containment arrangements to prevent the
escape of oil overboard. Before bunkering commences it is essential that the
Chief Engineer ensures that the bunkering plan is posted at the fuelling
station or control room and all crew members involved have a clear
understanding of their tasks and responsibilities.
ELETTRA - Basili >>
```

F.M.Zanotto, R.Basili

Abitilità Informatiche
Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Gestione Periferiche di I/O

- Un esempio: uso della redirectione in Unix (3)

ELETTRA - Basili >>cat sample.txt > copia.txt

ELETTRA - Basili >> ls -al

-rw-r--r--	1	basili	users	100	Sep	14	15:35	.login
-rw-r--r--	1	basili	users	254	Sep	14	15:35	.cshrc
drwxrwxrwx	2	basili	users	512	Sep	23	18:19	Japan#
drwxr-xr-x	2	basili	users	512	Nov	29	11:54	Lyras
drwxr-xr-x	2	basili	users	512	May	6	1999	Mail
-rw-r--r--	1	basili	users	0	Sep	14	15:35	dead.letter
-rw-rw-rw-	1	basili	users	1698	Dec	6	18:27	sample.txt
-rw-rw-rw-	1	basili	users	20088	Dec	6	18:26	sample.txt.mor
-rw-rw-rw-	1	basili	users	3554	Dec	6	18:33	sample.txt.tok
-rw-rw-rw-	1	basili	users	1698	Dec	15	17:55	copia.txt

ELETTRA - Basili >>

Abilità Informatiche

Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Gestione Periferiche di I/O

- Un esempio: uso della redirectione in Unix (4)

ELETTRA - Basili >>cat copia.txt

The Chief Engineer is personally responsible to supervise the bunkering operations at all times and for containment arrangements to prevent the escape of oil overboard. Before bunkering commences it is essential that the Chief Engineer ensures that the bunkering plan is posted at the fuelling station or control room and all crew members involved have a clear understanding of their tasks and responsibilities.

ELETTRA - Basili >>

Abilità Informatiche

Facoltà di Lettere e Filosofia

University of Naples "San Vito"



Gestione Periferiche di I/O

- Un esempio: uso dei file a video in Unix (5)

ELETTRA - Basili >>ls -al

drwxrwxrwx 2 basili users 512 Sep 23 18:19 Japan#

drwxr-xr-x 2 basili users 512 Nov 29 11:54 Lyras

drwxr-xr-x 2 basili users 512 May 6 1999 Mail

-rw-r--r-- 1 basili users 0 Sep 14 15:35 dead.letter

-rw-rw-rw- 1 basili users 1698 Dec 6 18:27 sample.txt

-rw-rw-rw- 1 basili users 20088 Dec 6 18:26 sample.txt.mor

-rw-rw-rw- 1 basili users 3554 Dec 6 18:33 sample.txt.tok

ELETTRA - Basili >>

Abilità Informatiche

Facoltà di Lettere e Filosofia

Università degli Studi "San Vito"

Università degli Studi "San Vito"

Logo

Logo

Cosa abbiamo visto

- Definizione di Sistema Operativo
- Macchina astratta
- Moltiplicazione risorse: macchine virtuali
- Moltiplicazione risorse: come otteniamo il miracolo?
 - Processo
 - Scheduling
- Gestione risorse
 - Memoria Centrale
 - Memoria di Massa
 - Dispositivi di I/O

F.M. Zuccherato, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia
