Sistemi Operativi e Reti di Calcolatori (SOReCa)

Corso di Laurea in *Ingegneria Informatica e Automatica (BIAR)*

Terzo Anno | Primo Semestre

A.A. 202425

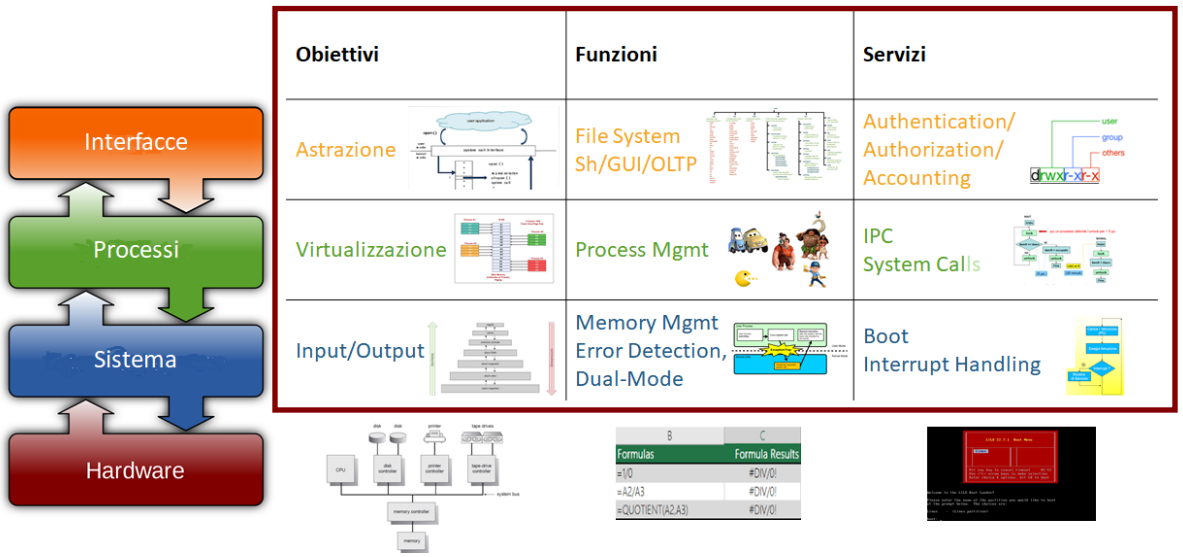
Esercizi – esempi di Test

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Sistemi operativi (3 CFU)

- Il sistema operativo
- Concorrenza e sincronizzazione
- Deadlock
- Inter-process communication (IPC)
- Scheduling
- Memoria centrale e virtuale
- Memoria di massa e File system
- Sicurezza informatica

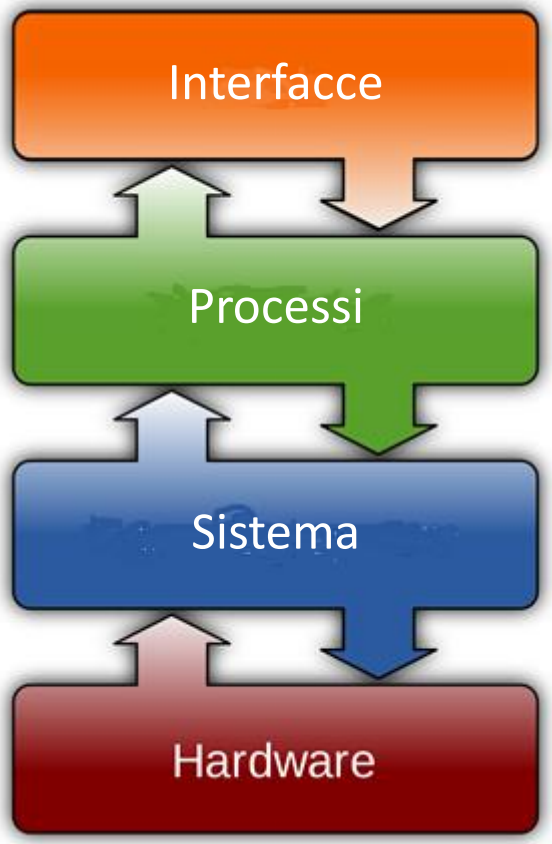


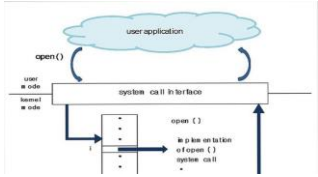
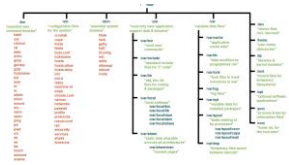
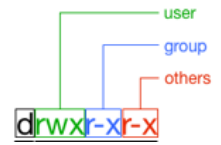
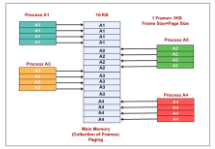
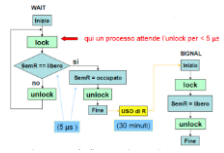
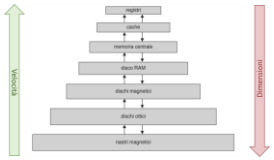
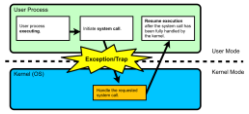
Lezioni: Settembre - Ottobre

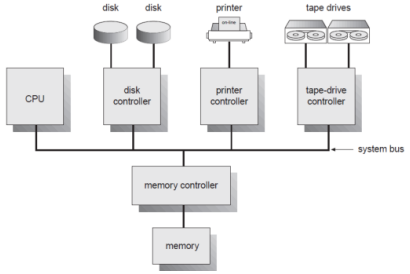
Dipartimento di Ingegneria informatica
automatica e gestionale Antonio Ruberti

Sapienza
Università di Roma

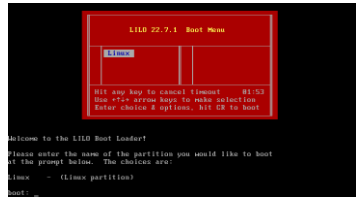
Operating Systems: 3x3 Matrix



	Obiettivi	Funzioni	Servizi
	Astrazione 	File System Sh/GUI/OLTP 	Authentication/ Authorization/ Accounting 
	Virtualizzazione 	Process Mgmt 	IPC System Calls 
	Input/Output 	Memory Mgmt Error Detection, Dual-Mode 	Boot Interrupt Handling 



B	C
Formulas	Formula Results
=1/0	#DIV/0!
=A2/A3	#DIV/0!
=QUOTIENT(A2,A3)	#DIV/0!



DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 1

Quale delle seguenti affermazioni sul kernel di un sistema operativo è vera?

A	È responsabile dell'accensione del computer
B	Viene swappato dal disco alla memoria principale ad ogni context switch
C	È responsabile, tra le altre cose, della gestione dei processori
D	Nessuna delle altre opzioni è corretta

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

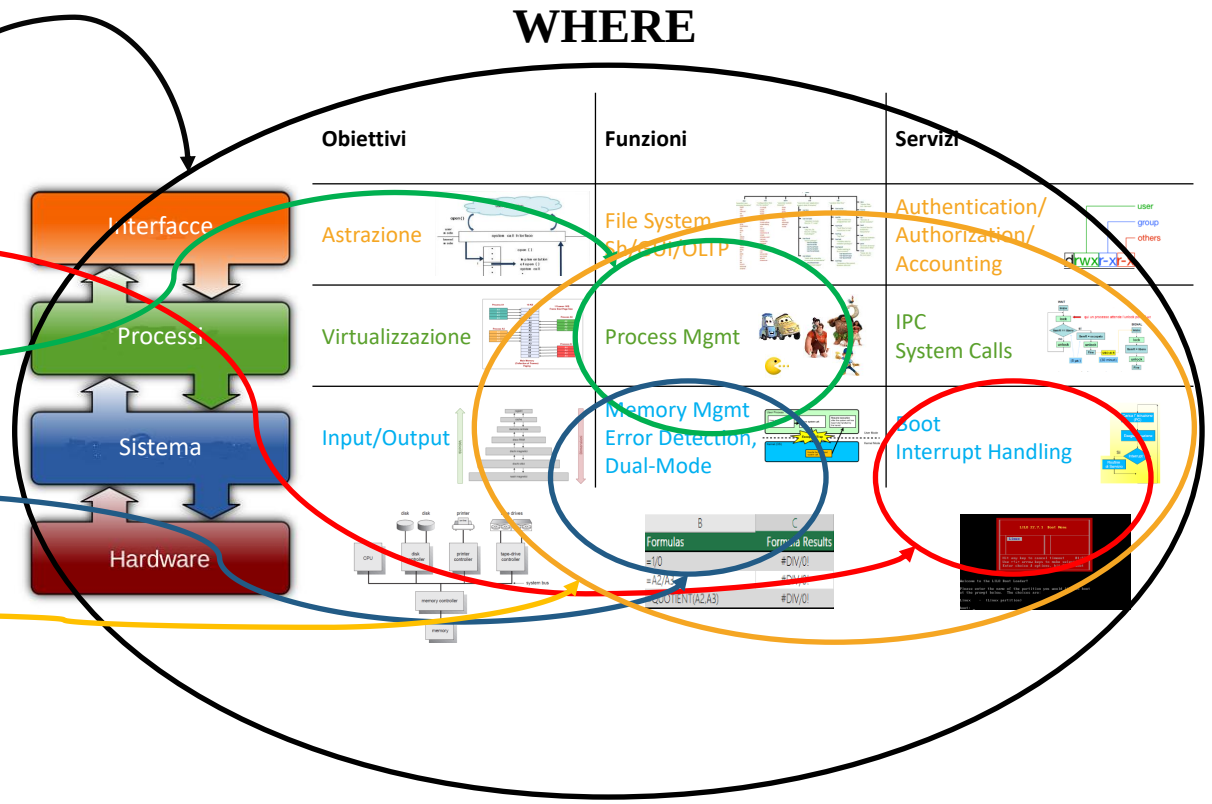
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 1b

Quale delle seguenti affermazioni sul kernel di un sistema operativo è vera?

A	È responsabile dell'accensione del computer
B	Viene swappato dal disco alla memoria principale ad ogni context switch
C	È responsabile, tra le altre cose, della gestione dei processori
D	Nessuna delle altre opzioni è corretta



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

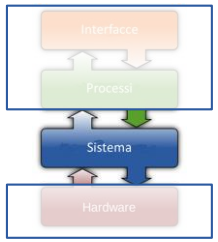
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 1b1

Operating Systems: Obiettivi Funzioni Servizi

Boot



I passi sono:

- Accensione
- Esecuzione del BIOS (in ROM o EEPROM) – *Basic I/O System*
  - Inizializza tutti i registri della CPU, i controllori delle periferiche e la memoria centrale
- Esecuzione del *bootstrap program o bootstrap loader*
  - caricamento del kernel del SO da una unità di memoria ed esecuzione
- Controllo del sistema da parte del SO

Permettere al sistema di ripartire in ogni momento.

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

A	È responsabile dell'accensione del computer

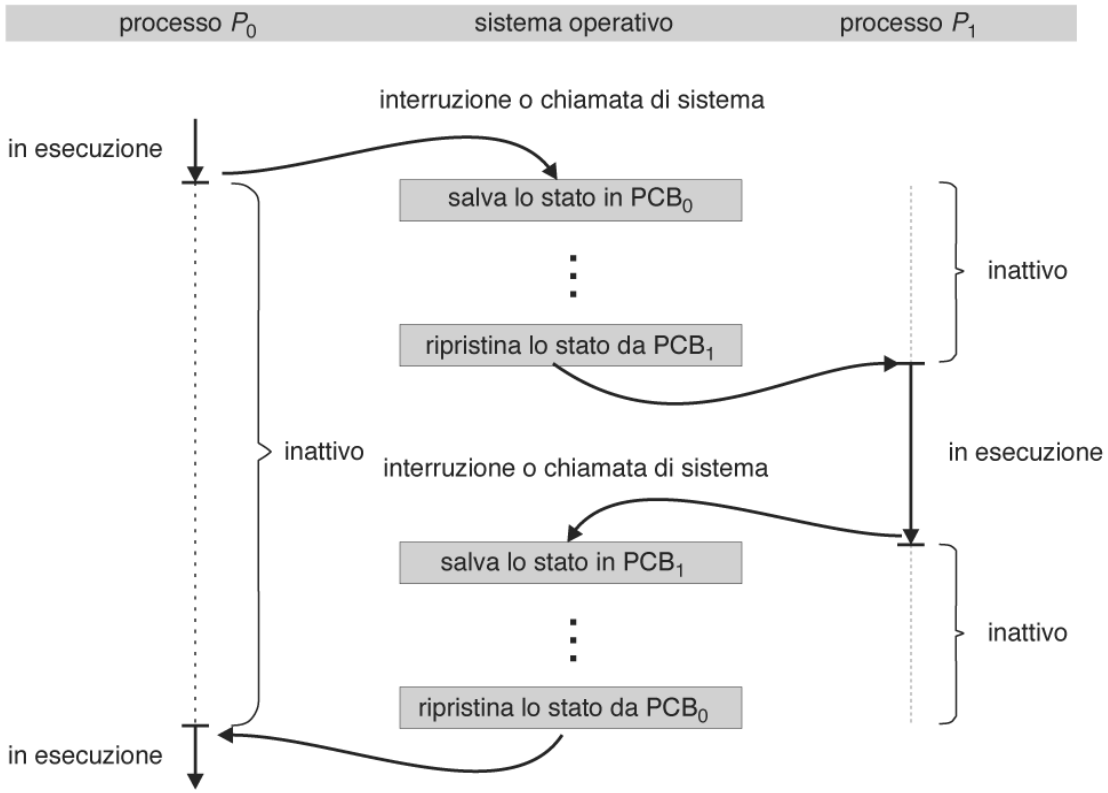
SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 1b2

Operating Systems: Processi

Context Switch



Context Switch (Cambio di Contesto): passaggio della CPU da un processo ad un altro

Context Switch =
sospensione del processo in esecuzione
+ caricamento del nuovo processo da mettere in esecuzione

- Il tempo per il cambio di contesto è puro tempo di gestione del sistema (non vengono compiute operazioni utili per la computazione dei processi)
- I tempi (10ms) per i cambi di contesto dipendono dal supporto hardware (e.g. registri disponibili e/o registri dedicati). <

B	Viene swappato dal disco alla memoria principale ad ogni context switch

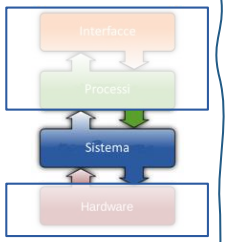
SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

Operating Systems: Esercizi

Domanda 1b3

Operating Systems: Obiettivi Funzioni Servizi

Dual-Mode



- Una volta avviato il sistema operativo, la CPU può operare in due modalità
  1. **User mode** – la CPU sta eseguendo codice di un utente
  2. **Kernel mode** (anche supervisor mode, system mode, monitor mode) – la CPU sta eseguendo codice del sistema operativo

- La CPU ha un **Mode bit** nel *registro di stato* (Process Status Word o PSW) che indica in quale modo si trova: kernel (0) o user (1)

Rendere il sistema quanto più possibile scevro da esecuzioni pericolose.

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

C	È responsabile, tra le altre cose, della gestione dei processori

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 1c

Quale delle seguenti affermazioni sul kernel di un sistema operativo è vera?

A	È responsabile dell'accensione del computer
B	Viene swappato dal disco alla memoria principale ad ogni context switch
C	È responsabile, tra le altre cose, della gestione dei processori
D	Nessuna delle altre opzioni è corretta

WHEN	WHO	WHAT	HOW
After Booting	SO Kernel	-	-
Booting	BIOS	System	Checking
Process Management, Scheduling	SO Kernel	Process	Long Term Scheduling
Process Management	SO Kernel	Resource Allocation	Scheduling
!passpartout	N/A	N/A	N/A

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 1d

Quale delle seguenti affermazioni sul kernel di un sistema operativo è vera?

A	~~È responsabile dell'accensione del computer~~
B	~~Viene swappato dal disco alla memoria principale ad ogni context switch~~
C	È responsabile, tra le altre cose, della gestione dei processori
D	~~Nessuna delle altre opzioni è corretta~~

WHY

	Booting ➔ BIOS
	Kernel should not be swapped out
	Resource Allocation
	C corretta

Operating Systems: Esercizi

Domanda 2

Quale dei seguenti elementi fa parte del process control block?

A	Nessuna delle altre opzioni contiene elementi del process control block
B	Le informazioni sul contesto del processo, aggiornate ad ogni istruzione eseguita
C	L'intera immagine del processo in memoria
D	La tabella delle pagine di secondo livello

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

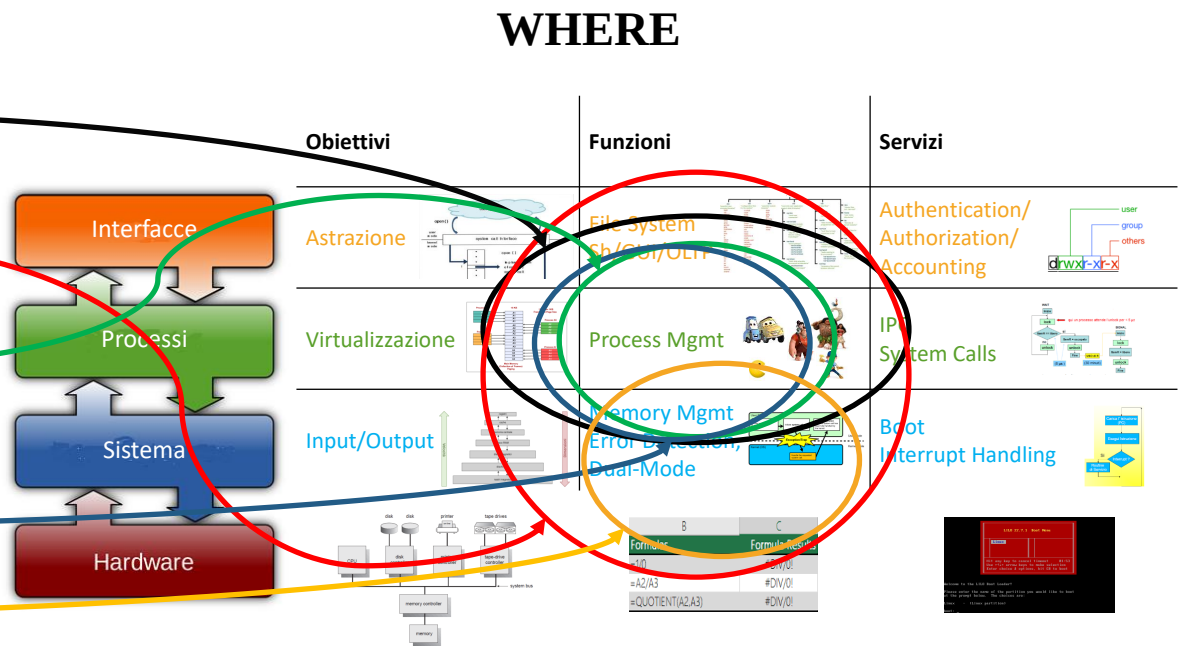
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 2b

Quale dei seguenti elementi fa parte del process control block?

A	Nessuna delle altre opzioni contiene elementi del process control block
B	Le informazioni sul contesto del processo, aggiornate ad ogni istruzione eseguita
C	L'intera immagine del processo in memoria
D	La tabella delle pagine di secondo livello



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

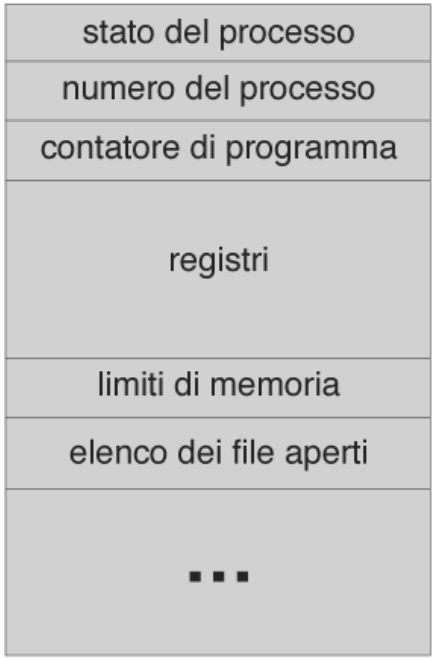
Domanda 2b1

Operating Systems: Processi

PCB

Process Control Block (PCB): Struttura dati del **kernel** che mantiene le informazioni sul processo

- Stato del processo
- Identificatore del processo (Numero)
- Program counter
- i.e. indirizzo istruzione successiva
- Registri della CPU
- Stack Pointer
- Frame Pointer
- I valori devono essere salvati per poter riprendere correttamente l'esecuzione dopo un interruzione
- Informazioni sullo scheduling (es. priorità, etc …)
- Informazioni sulla gestione della memoria (es. tabelle delle pagine, etc …)
- Informazioni di contabilizzazione delle risorse (es. tempo uso CPU, etc …)
- Informazioni sullo stato dell'I/O (es. lista dispositivi assegnati al processo, elenco file aperti, etc …)



SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

B | Le informazioni sul contesto del processo, aggiornate ad ogni istruzione eseguita

SAPIENZA UNIVERSITÀ DI ROMA

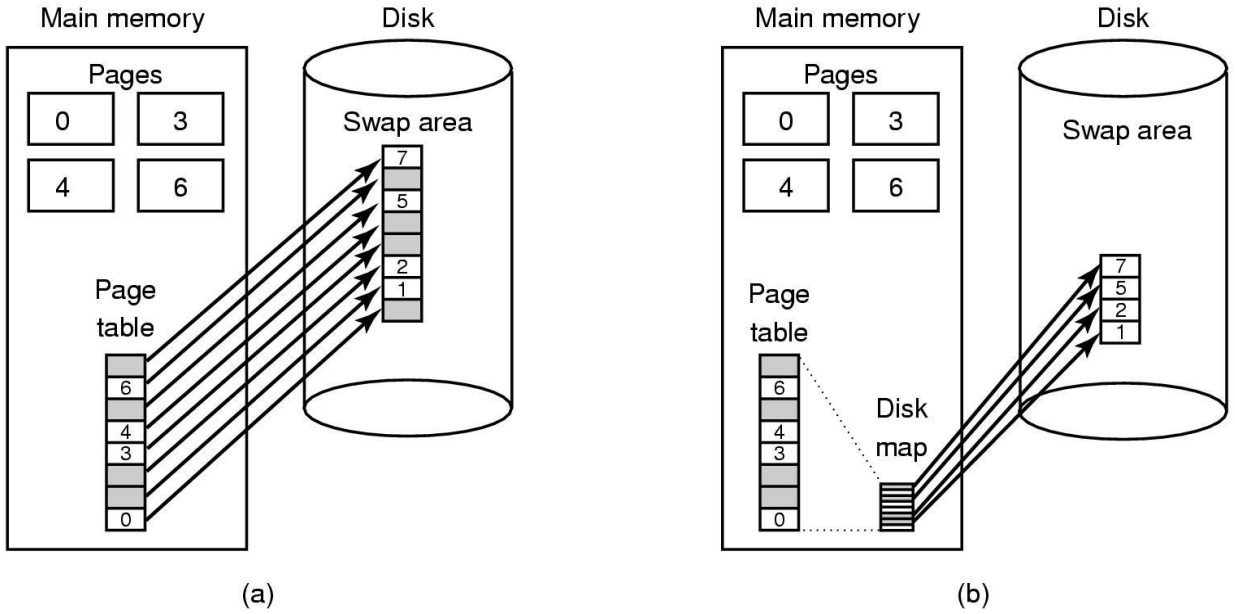
Operating Systems: Esercizi

Domanda 2b2

Operating Systems: Memory Management

Virtual Memory: Backing Store 1/2

Memory Manager: gestione della movimentazione delle pagine da e verso il disco.



(a) **Dynamic**: alloca le pagine secndo necessità, a *run-time*. Non assegna per il processo in anticipo lo spazio su disco. Necessita di un **Disk Map**

Backing Store

Gestione dell'area di swap su disco:

- Partizione separata senza file system
- File senza struttura

Allocazione spazio di swap:

(a) **Static**: Assegna una partizione fissa (chunk) abbastanza grande al processo all'avvio.

Lista di chunk liberi.

Indirizzo virtuale su disco:

- Indirizzo iniziale della partizione.
- offset di pagina.

Aree diverse per:

- Dati (cresce)
- Testo (non cresce)
- Stack (cresce)

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

C	L'intera immagine del processo in memoria

SAPIENZA UNIVERSITÀ DI ROMA

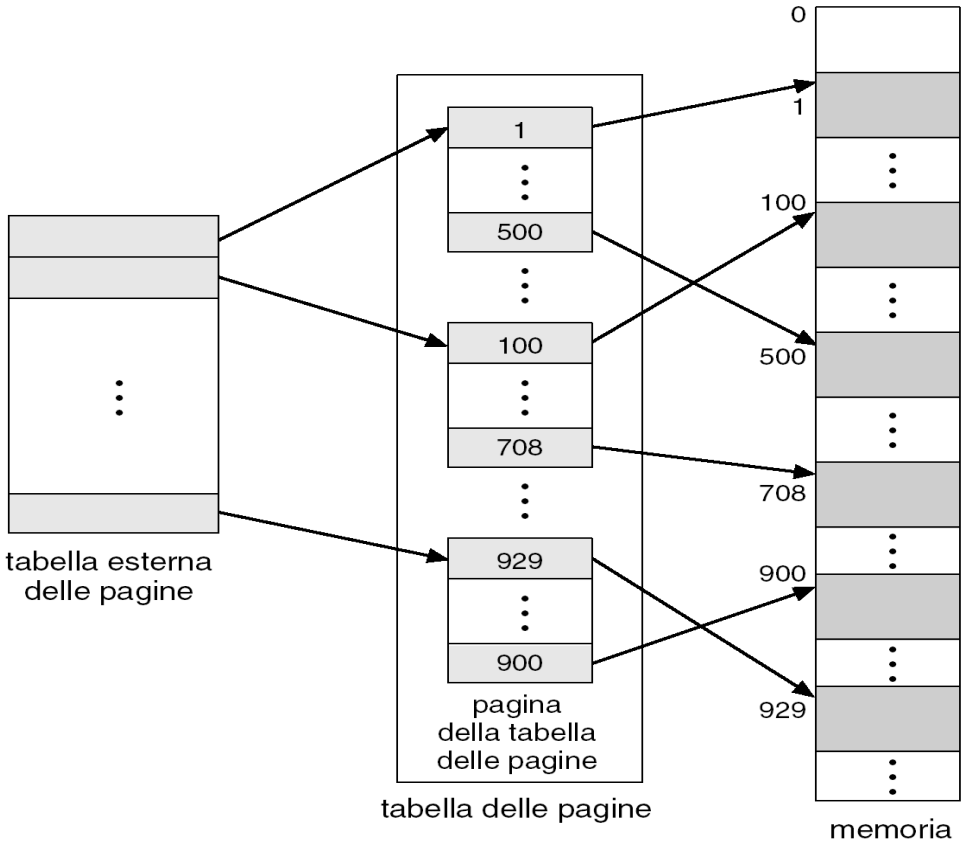
Operating Systems: Esercizi

Domanda 2b3

Operating Systems: Memory Management

Virtual Memory: Paging 7c/7

Memory Manager: strutture avanzate per la tabella delle pagine.



Paginazione Gerarchica

- **Soluzione gerarchica**: suddividere lo spazio degli indirizzi logici in più tabelle di pagine (es. 2 Livelli)
- Una directory delle pagine detta tabella esterna: ogni elemento punta ad una sotto-tabella delle pagine
- Un insieme di sotto-tabelle delle pagine (Tipicamente ampia quanto una pagina al fine di essere completamente contenuta in un frame)

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

D	La tabella delle pagine di secondo livello

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 2c

Quale dei seguenti elementi fa parte del process control block?

A	Nessuna delle altre opzioni contiene elementi del process control block
B	Le informazioni sul contesto del processo, aggiornate ad ogni istruzione eseguita
C	L'intera immagine del processo in memoria
D	La tabella delle pagine di secondo livello

WHEN	WHO	WHAT	HOW
Context Switch	Short Term Scheduling	Process Status	
!passpartout	N/A	N/A	N/A
Ad ogni istruzione eseguita?!?	-	-	-
Memory Management	-	-	-
Memory Management	-	-	-

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 2d

Quale dei seguenti elementi fa parte del process control block?

A	Nessuna delle altre opzioni contiene elementi del process control block
B	~~Le informazioni sul contesto del processo, aggiornate ad ogni istruzione eseguita~~
C	~~L'intera immagine del processo in memoria~~
D	~~La tabella delle pagine di secondo livello~~

WHY

	Nessuna delle altre opzioni è corretta
	Le informazioni **non** sono aggiornate ad **ogni istruzione eseguita** ma solo prima del context switch
	Inerente il **Memory Management**
	Inerente il **Memory Management**

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 3

Quale delle seguenti affermazioni sul controllo di accesso è vera?

A	Nel controllo di accesso basato su ruoli, ad ogni ruolo è assegnato un utente
B	Nessuna delle altre opzioni è vera
C	Nel controllo di accesso basato su ruoli, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-ruoli-oggetti
D	Nel controllo di accesso discrezionale, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-oggetti

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

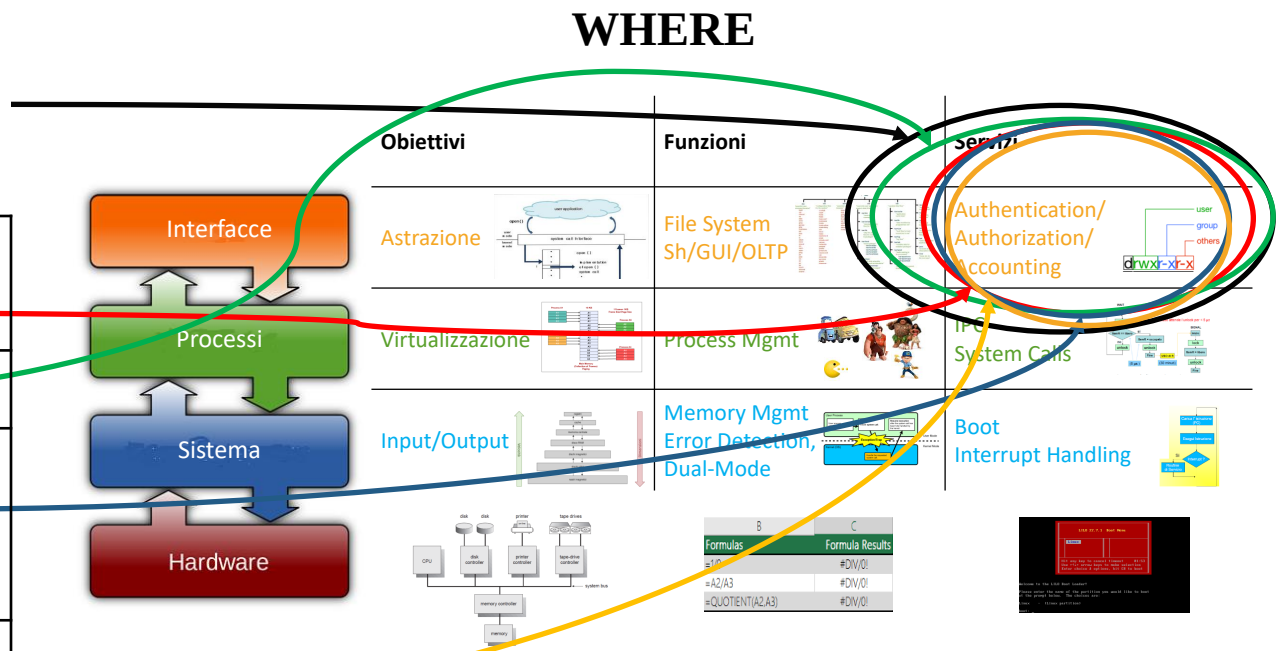
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 3b

Quale delle seguenti affermazioni sul controllo di accesso è vera?

A	Nel controllo di accesso basato su ruoli, ad ogni ruolo è assegnato un utente
B	Nessuna delle altre opzioni è vera
C	Nel controllo di accesso basato su ruoli, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-ruoli-oggetti
D	Nel controllo di accesso discrezionale, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-oggetti



DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Operating Systems: Access Control

Domanda 3b: Tipi di Access Control

MAC: Mandatory Access Control ➔ security policy: security labels

R(o)BAC: Role Based Access Control ➔ Role != Group

RBAC: Rule Based Access Control ➔ ACL: Access Control List

DAC: Discretionary Access Control ➔ ACL: Access Control List

Cfr. «**08-SOReCa-Sicurezza**»

Dipartimento di Ingegneria informatica
automatica e gestionale Antonio Ruberti

Sapienza
Università di Roma

Operating Systems: Access Control

Domanda 3b: MAC

MAC: Mandatory Access Control ➔ security policy: security labels

Operating Systems: Security & Protection

Confidentiality ➔ Bell-La Padula Model 3/3

Cosa facciamo
- L'intelligence
- Collaborazione istituzionale
- Rapporti con l'Autorità giudiziaria
- Tutela delle informazioni
  - Autorità nazionale per la sicurezza
  - Il segreto di Stato
  - Classifiche di segretezza
  - Rilascio delle abilitazioni di sicurezza
- I controlli sul Sistema

Home » Cosa facciamo » Tutela delle informazioni » Classifiche di segretezza

Classifiche di segretezza

La classifica di segretezza è l'**indicatore del livello di segretezza** attribuito in ambito nazionale a una determinata informazione. Si configurano come **documenti classificati** qualsiasi supporto – materiale o immateriale, analogico o digitale – contenente informazioni classificate e, pertanto, sottoposto a misure di protezione fisica, logica e tecnica dal momento della sua origine fino a quello della sua distruzione o declassifica. Durante tale arco di vita, la sua trattazione e gestione sono disciplinate da modalità specifiche. Le singole parti di un documento possono richiedere classifiche differenti. In questo caso il livello generale di classifica dell'intero documento è pari almeno a quello della parte con classifica più elevata.

Le classifiche sono quattro:

- **segretissimo** (SS)
- **segreto** (S)
- **riservatissimo** (RR)
- **riservato** (R)

Rep. Italiana	NATO
Segretissimo (**SS**)	Top Secret
Segreto (**S**)	Secret
Riservatissimo (**RR**)	Confidential
Riservato (**R**)	Reserved

➔ Nulla Osta di Sicurezza (NOS) ➔ Livello (R, RR, S, SS)

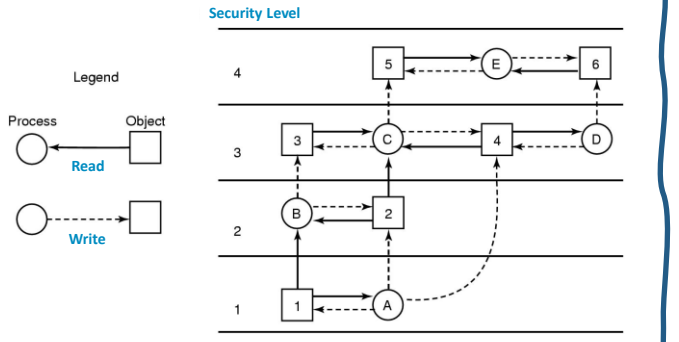
SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

Confidentiality ➔ Bell-La Padula Model 2/3

Modello Bell-La Padula: 2 regole (proprietà)

1. **No Read Up** (Simple Security Property ➔ non leggere informazioni potenzialmente più confidenziali
2. **No Write Down** (* Property) ➔ non scrivere inavvertitamente informazioni più confidenziali



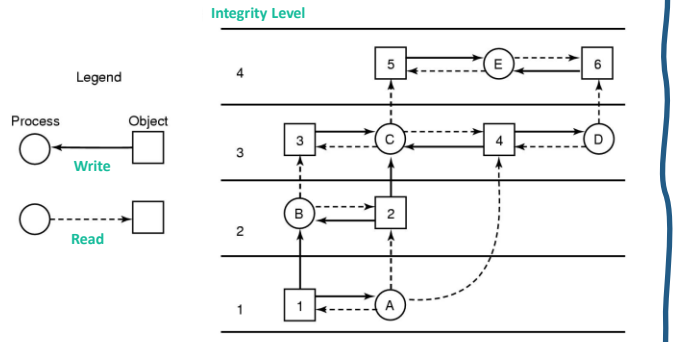
SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

Integrity ➔ Biba Model 2/2

Modello Biba: 2 regole (proprietà)

1. **No Write Up** (Simple Integrity Principle) ➔ non inserire informazioni meno integre
2. **No Read Down** (Integrity * Property) ➔ non utilizzare informazioni meno integre



SAPIENZA UNIVERSITÀ DI ROMA

AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

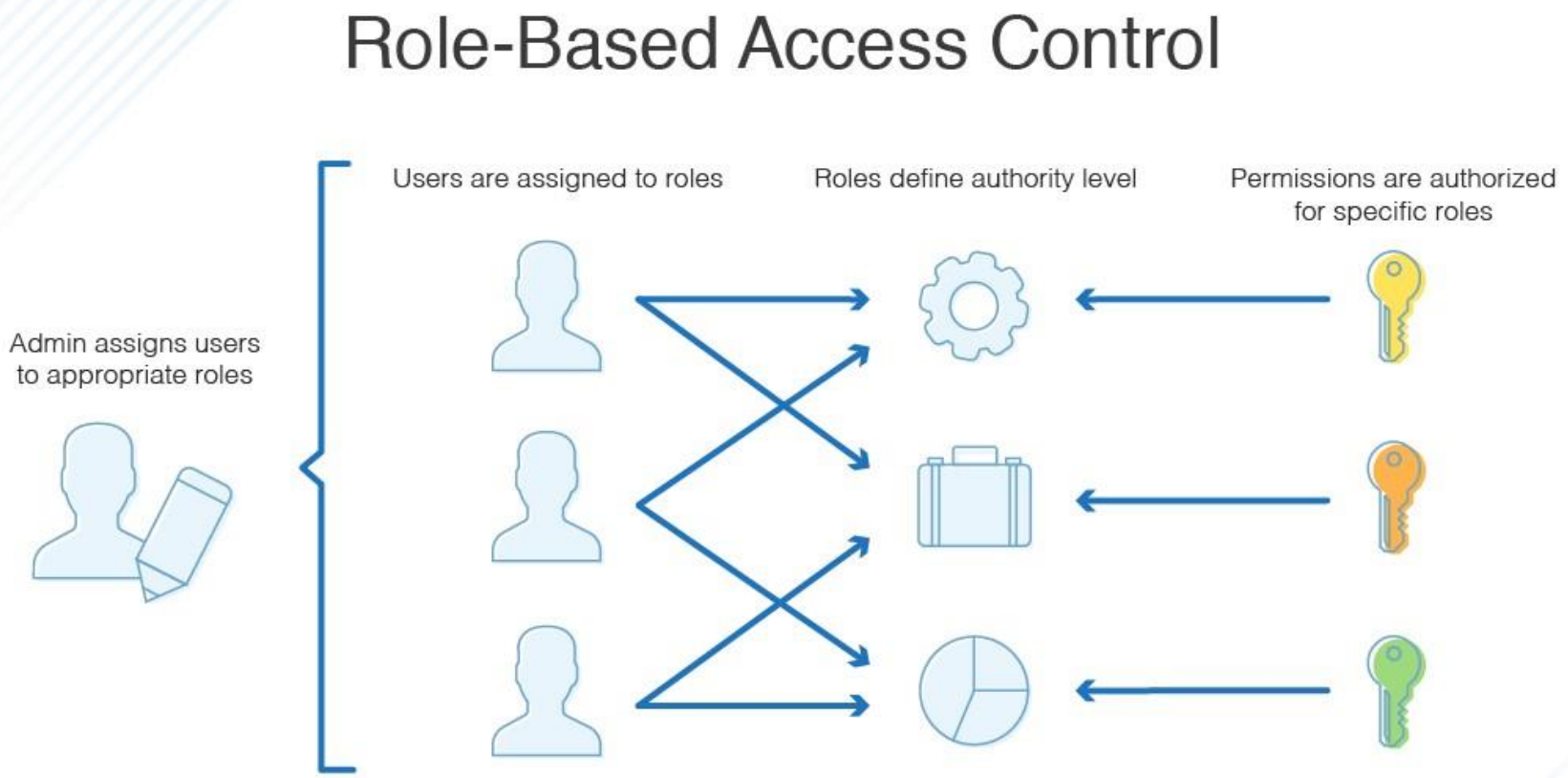
Operating Systems: Access Control

Domanda 3b: RBAC

RBAC: Role Based Access Control ➔ Role != Group

Implementato tramite Directory Server (LDAP): DB gerarchico

es. OpenLDAP, MS-AD (Microsoft Active Directory)



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Access Control

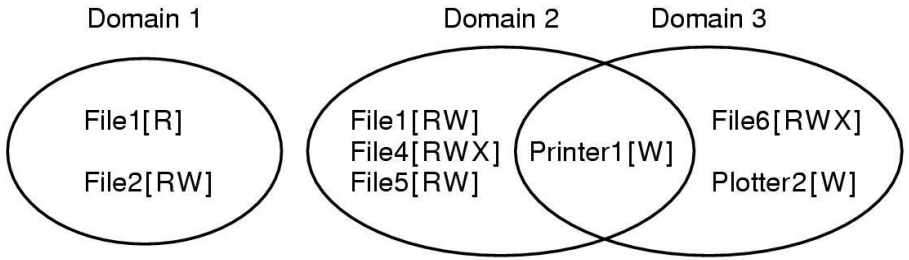
Domanda 3b: R(u)BAC

R(u)BAC:
Rule Based Access Control
➔ ACL:
Access Control List

Operating Systems: Security & Protection

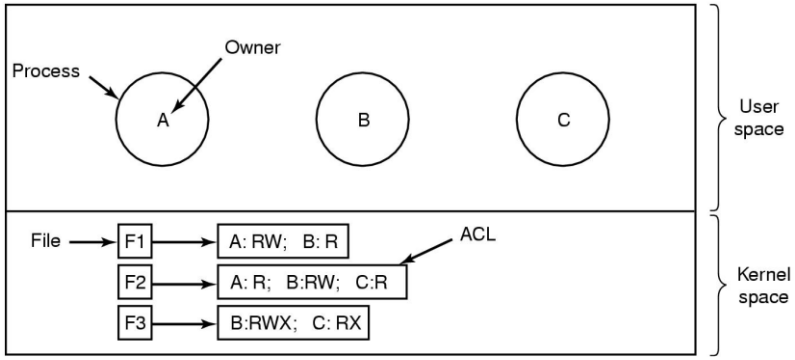
AAA 6/8

Permission: autorizzazione all'accesso mediante regole.



Domini: insiemi di risorse per organizzarne l'accesso

Protection Domain: …



Domain \ Object	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

ACL: Access Control List

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Access Control

Domanda 3b: DAC

DAC: Discretionary Access Control ➔ ACL: Access Control List

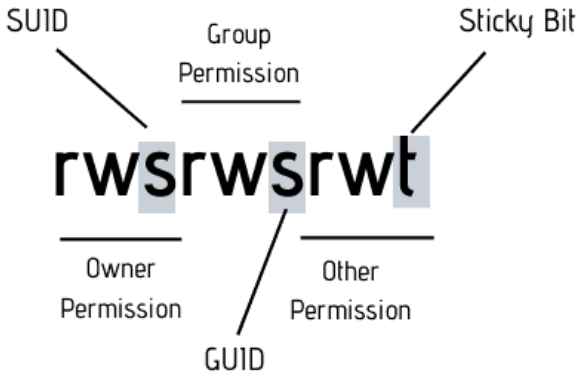
Operating Systems: Security & Protection

Protezione 1/6

ACL ed altri permessi: Access Control List

ACL: permessi inclusi in una lista

File	Access control list
Password	tana, sysadm: RW
Pigeon_data	bill, pigfan: RW; tana, pigfan: RW; ...



Modifiche ai normali permessi di esecuzione (x):
SUID (s): Set User ID. Indica che il file va eseguito con i privilegi dell'utente proprietario del file anziché con quelli dell'utente che lo avvia. Non si applica alle directory.
GUID (s): Set Group ID. indice che il file va eseguito con i permessi del gruppo assegnato al file anziché quelli del gruppo principale dell'utente che lo avvia. Non si applica alle directory.
Sticky (t): applicato ai file eseguibili, suggerisce al kernel di mantenere nel file di swap una copia del file eseguibile anche dopo che era terminato. Applicato alle directory, i file in essa contenuti possono essere cancellati e spostati solamente dagli utenti che ne sono proprietari, o dall'utente proprietario della directory che li contiene, o ancora dal superuser

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 3c

Quale delle seguenti affermazioni sul controllo di accesso è vera?

A	Nel controllo di accesso basato su ruoli, ad ogni ruolo è assegnato un utente
B	Nessuna delle altre opzioni è vera
C	Nel controllo di accesso basato su ruoli, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-ruoli-oggetti
D	Nel controllo di accesso discrezionale, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-oggetti

WHEN	WHO	WHAT	HOW
Access Control	Authorization	Process Status	MAC DAC RBAC
RBAC	Utenti	Ruolo Risorsa	Utente ∈ Role
!passpartout	-	-	-
RBAC	Soggetto (Utente)	Ruolo Oggetto	Tabelle: User-Role Role-Obj
DAC	Soggetto (Utente)	Oggetto	tabella

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 3d

WHY

Quale delle seguenti affermazioni sul controllo di accesso è vera?

A	~~Nel controllo di accesso basato su ruoli, ad ogni ruolo è assegnato un utente~~	Possono esistere ruoli senza utenti
B	~~Nessuna delle altre opzioni è vera~~	Opzione D possibile
C	~~Nel controllo di accesso basato su ruoli, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-ruoli-oggetti~~	Vi sono 2 tabelle: Soggetti-Ruoli Ruoli-Oggetti
D	Nel controllo di accesso discrezionale, prima di stabilire se un'operazione è lecita, è necessario consultare una tabella soggetti-oggetti	Per accedere ad un oggetto Tabella soggetti-oggetti (es. **File System Permission**)

	Object										
Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

Operating Systems: Esercizi

Domanda 4

Quale delle seguenti affermazioni, riguardanti il joint progress diagram di 2 processi, è vera?

A	Nessuna delle altre opzioni è vera
B	Può essere usato per visualizzare le possibilità di deadlock, ma solo se i processi richiedono al massimo 2 risorse
C	Può essere usato per determinare quando uno dei due processi va in esecuzione a discapito dell'altro
D	Può essere usato per determinare quando uno dei due processi sperimenta un page fault

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

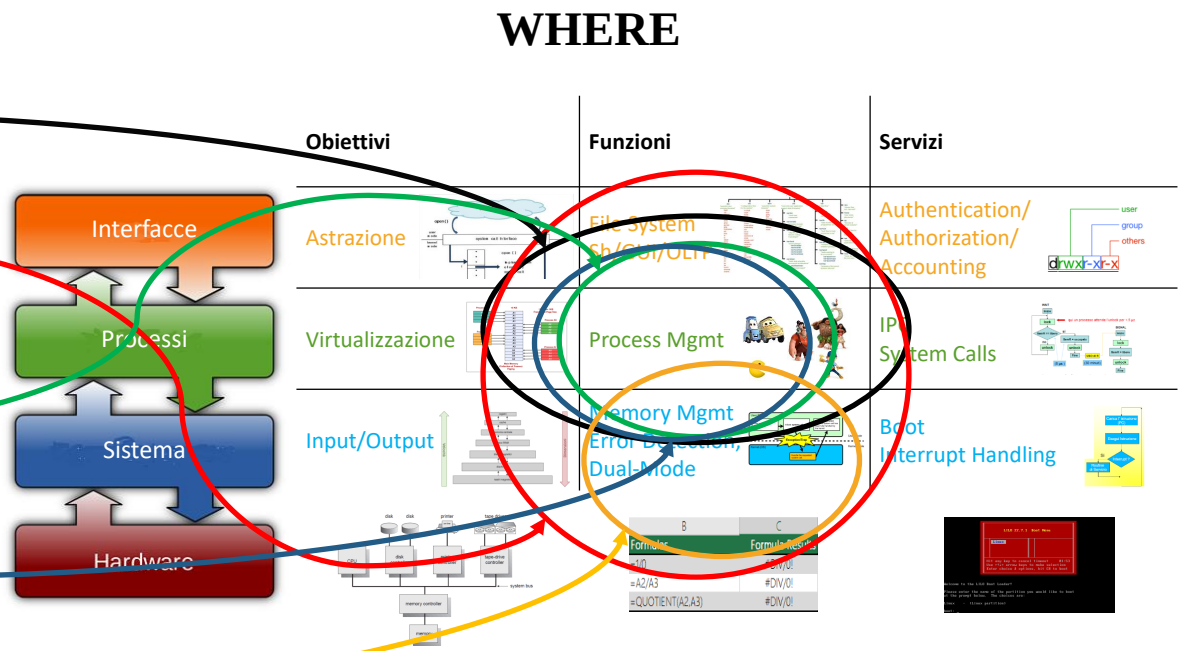
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 4b

Quale delle seguenti affermazioni, riguardanti il joint progress diagram di 2 processi, è vera?

A	Nessuna delle altre opzioni è vera
B	Può essere usato per visualizzare le possibilità di deadlock, ma solo se i processi richiedono al massimo 2 risorse
C	Può essere usato per determinare quando uno dei due processi va in esecuzione a discapito dell'altro
D	Può essere usato per determinare quando uno dei due processi sperimenta un page fault



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

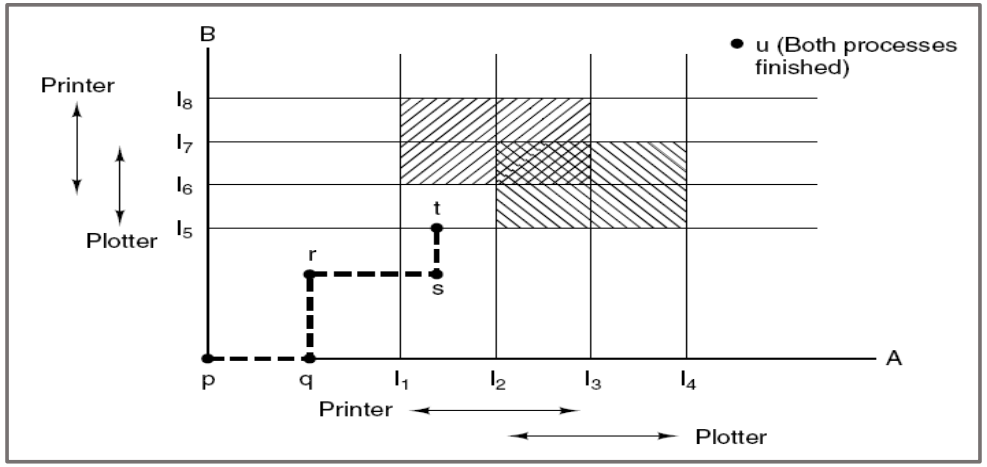
Domanda 4b1

Operating Systems: Deadlock

Deadlock: Evitare 3/7

Grafo di Allocazione Ogni punto del diagramma rappresenta uno stato congiunto dei due processi.

- p: nessun processo eseguito
- q: A eseguito, B comincia
- r: A ricomincia, in I_1 chiede il plotter
- …



modello per gestire due processi e due risorse, ad esempio una stampante e un plotter. L'asse orizzontale rappresenta il numero di istruzioni eseguite dal processo A. L'asse verticale rappresenta il numero di istruzioni eseguite dal processo B. In I_1 A richiede una stampante; a I_2 ha bisogno di un plotter. La stampante e il plotter vengono rilasciati rispettivamente in I_3 e I_4. Il processo B necessita del plotter da I_5 a I_7 e della stampante da I_6 a I_8 .

Il **massimo** è relativo al **numero di processi**, non di risorse

B	Può essere usato per visualizzare le possibilità di deadlock, ma solo se i processi richiedono al massimo 2 risorse
C	Può essere usato per determinare quando uno dei due processi va in esecuzione a discapito dell'altro

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

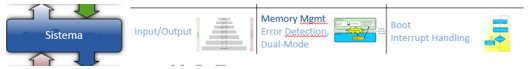
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 4b2

Operating Systems: Memory Management

Virtual Memory: Working Set 4/4



Memory Manager: riepilogo dei Page Replacement Algorithm.

Algorithm	Comment
Optimal	Not implementable, but useful as a benchmark
NRU (Not Recently Used)	Very crude approximation of LRU
FIFO (First-In, First-Out)	Might throw out important pages
Second chance	Big improvement over FIFO
Clock	Realistic
LRU (Least Recently Used)	Excellent, but difficult to implement exactly
NFU (Not Frequently Used)	Fairly crude approximation to LRU
Aging	Efficient algorithm that approximates LRU well
Working set	Somewhat expensive to implement
WSClock	Good efficient algorithm

Stack Algorithm

Algoritmi di Page Replacement per cui vale la disuguaglianza:

$M(m, r) \subseteq M(m+1, r)$

dove:

- **M**: insieme delle pagine del processo in memoria
- **m**: numero di pagine del processo in memoria
- **r**: indice scelta solo all'interno del processo con Page Fault

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

D	Può essere usato per determinare quando uno dei due processi sperimenta un page fault

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 4c

Quale delle seguenti affermazioni, riguardanti il joint progress diagram di 2 processi, è vera?

A	Nessuna delle altre opzioni è vera
B	Può essere usato per visualizzare le possibilità di deadlock, ma solo se i processi richiedono al massimo 2 risorse
C	Può essere usato per determinare quando uno dei due processi va in esecuzione a discapito dell'altro
D	Può essere usato per determinare quando uno dei due processi sperimenta un page fault

WHEN	WHO	WHAT	HOW
Avoid Deadlock	2 processi	Stato di allocazione delle risorse	Verifica a priori
!passpartout	N/A	N/A	N/A
Avoid deadlock	-	2 risorse	-
Avoid Deadlock	2 processi	esecuzione	A priori
Memory Management	2 processi	Page fault	Working Set

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 4d

WHY

Quale delle seguenti affermazioni, riguardanti il joint progress diagram di 2 processi, è vera?

A	~~Nessuna delle altre opzioni è vera~~
B	~~Può essere usato per visualizzare le possibilità di deadlock, ma solo se i processi richiedono al massimo 2 risorse~~
C	Può essere usato per determinare quando uno dei due processi va in esecuzione a discapito dell'altro
D	~~Può essere usato per determinare quando uno dei due processi sperimenta un page fault~~

	Opzione C corretta
	Il massimo è relativo al numero di **processi**, non di risorse
	Ideato per evitare i deadlock, utile anche per visualizzare quando viene eseguito un processo o l'altro
	Inerente il **Memory Management**

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 5

Quale dei seguenti elementi non è una delle parti che definiscono un processo?

A	Informazioni sullo stato delle risorse
B	La priorità
C	Il contatore di programma
D	I dati contenuti nella porzione di memoria a lui dedicata

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

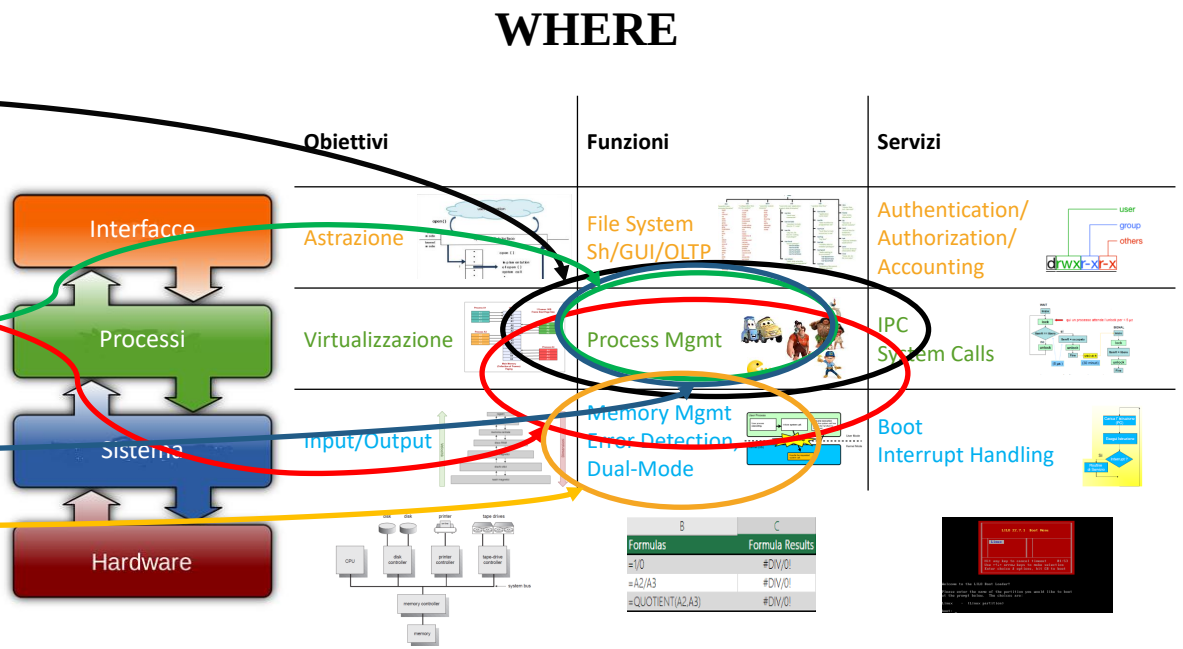
Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 5b

Quale dei seguenti elementi **non** è una delle parti che definiscono un processo?

A	Informazioni sullo stato delle risorse
B	La priorità
C	Il contatore di programma
D	I dati contenuti nella porzione di memoria a lui dedicata



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

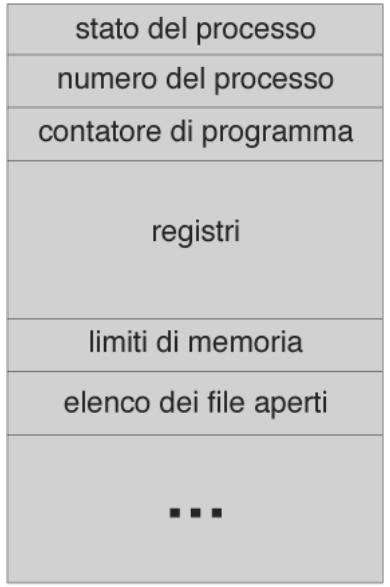
Domanda 5b1

Operating Systems: Processi

PCB

Process Control Block (PCB): Struttura dati del **kernel** che mantiene le informazioni sul processo

- Stato del processo
- Identificatore del processo (Numero)
- Program counter
- i.e. indirizzo istruzione successiva
- Registri della CPU
- Stack Pointer
- Frame Pointer
- I valori devono essere salvati per poter riprendere correttamente l'esecuzione dopo un interruzione
- Informazioni sullo scheduling (es. priorità, etc …)
- Informazioni sulla gestione della memoria (es. tabelle delle pagine, etc …)
- Informazioni di contabilizzazione delle risorse (es. tempo uso CPU, etc …)
- Informazioni sullo stato dell'I/O (es. lista dispositivi assegnati al processo, elenco file aperti, etc …)



SAPIENZA UNIVERSITÀ DI ROMA

A	Informazioni sullo stato delle risorse
B	La priorità
C	Il contatore di programma

Operating Systems: Esercizi

Domanda 5d

Quale dei seguenti elementi **non** è una delle parti che definiscono un processo?

A	~~Informazioni sullo stato delle risorse~~
B	~~La priorità~~
C	~~Il contatore di programma~~
D	I dati contenuti nella porzione di memoria a lui dedicata

WHY

	Nel PCB le risorse assegnate al processo
	Nel PCB: scheduling ed informazioni di stato
	Nel PCB: Il contatore di programma
	I dati contenuti nella porzione di memoria a lui dedicata

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Considerazioni Generali

Test a risposta multipla

Operating Systems: Esercizi

Tipi di Quiz

Comprensione totale solo se si è in grado di capire come è stato ideato ogni singolo quiz (identificare il processo mentale di chi ha creato la domanda per non cadere in facili trabocchetti). 2 direzioni di indagine:

- **Formulazione** della Domanda (**Stem**)
- **Scelta** delle **4** possibili **Risposte** (**1 Key, 3 Options**)

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Stem: Formulazione dei Quiz

La formulazione della domanda si effettua in genere in questo modo:

- **What-Is**: chiara e semplice, non ci sono tranelli scegliere la risposta giusta
- **What-Is-Not**: comprende 1 o più negazioni, per confondere (ricontare le negazioni presenti)
- **The-Best/Most**: insidiosa, individuare la risposta migliore

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Options: Scelta della Key

Le risposte vengono scelte in genere in questo modo:

- **Banale**: 1 sola risposta giusta, altre palesemente errate (es. espressioni appositamente coniate)
- **2-Bad**: 2 risposte palesemente non corrette → scegliere fra le altre 2 (risposta quiz “a 2 passate”)
- **Confusion**: tutte le risposte corrispondono a concetti dell'esame ma solo uno inerente la domanda
- **Letio-Difficiliora**: le risposte sono molto simili e differiscono per una espressione da ricordare

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Quiz: Griglia Complessiva

	What-Is	What-Is-Not	Best/Most
Banale	37%	16%	
2-Bad	13%	9%	
Confusion	8%	7%	
Letio Difficiliora	7%	3%	

Complessità

Complessità plausibile degli esami a quiz (non solo “Sistemi Operativi”!!!)

Punti di Attenzione:

•**Livelli**: difficolta diverse. Le Domanda banali richiedono meno tempo per essere risposte.

•**Passate**: prevedere 2 o 3 “passate”. Ad esempio:

1. Banali
2. 2-Bad
3. Confusion – Letio Difficiliora

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Quiz: Regole

Punti di Attenzione:

•**Copie Cartacee**: usare solo se non si deve effettuare una ricerca ➔ reperimento informazione <= 15 secondi ➔ non usare durante la prima passata

•**Punteggio**: molto penalizzata la risposta sbagliata (0,5 esatta) ➔ rispondere “a sorte” solo si è nella situazione **2-Bad**

Regole per gli Esami 3/9

Scritto: Le domande verteranno sempre sull’intero programma del corso

Si tratta di un compito a quiz da fare direttamente al computer

Ci saranno 25 domande da fare in 30 minuti

Per ogni domanda, ci sono 4 opzioni, delle quali una sola è giusta

Tutti gli studenti hanno lo stesso compito, ma con le domande (e le opzioni) mischiate

E possibile consultare copie cartacee di libro, slide, appunti.

Non è consentito, ovviamente, comunicare con altri studenti, né consultare archivi elettronici

Per ciascuno viene calcolato il punteggio come 2E − S, con E numero di risposte esatte ed S numero di risposte sbagliate quindi il punteggio va da -25 a 50

Per assegnare un voto da insufficiente a 22 a ciascun punteggio, si valuteranno i punteggi di tutti i partecipanti all’esame, seguendo (cum grano salis) una distribuzione a campana di Gauss

Non è necessario rispondere a tutte le domande

Chi fa il compito migliore rispetto agli altri ha il punteggio più alto

•**Tempo Medio**: 72 secondi (1 minuto e 12 secondi) a domanda.
Nel caso di 3 passate: 15 sec – 1 minuto – 2 minuti (esempio)

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 6

Quale delle seguenti affermazioni è falsa?

A	Nel caso di un sistema operativo a kernel separato, la gestione dei process switch è a sua volta un processo
B	Se le funzioni del sistema operativo vengono eseguite come processi separati, c'è sempre bisogno di un process switch per eseguire una funzionalità del sistema operativo
C	Se le funzioni del sistema operativo vengono eseguite all'interno dei processi utente, se un processo effettua una syscall e poi può continuare ad essere eseguito, non avviene alcun process switch
D	Se le funzioni del sistema operativo vengono eseguite all'interno dei processi utente, non c'è bisogno di un process switch per eseguire una funzionalità del sistema operativo

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

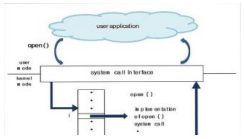
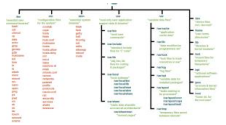
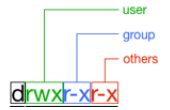
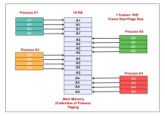
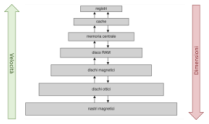
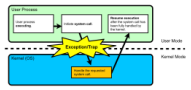
Domanda 6 b

«**3. Strutture dei Sistemi Operativi**»

Operating Systems: Organizzazione

Tipologie di Sistemi Operativi

- Monolitico
- Stratificato
- Microkernel
- Modulari
- A macchine virtuali
- Client/Server

Obiettivi	Funzioni	Servizi
Astrazione 	File System Sh/GUI/OLTP 	Authentication/ Authorization/ Accounting 
Virtualizzazione 	Process Mgmt 	IPC System Calls 
Input/Output 	Memory Mgmt Error Detection, Dual-Mode 	Boot Interrupt Handling 

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

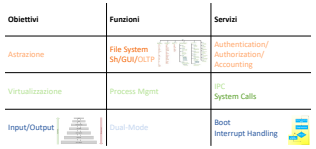
Operating Systems: Esercizi

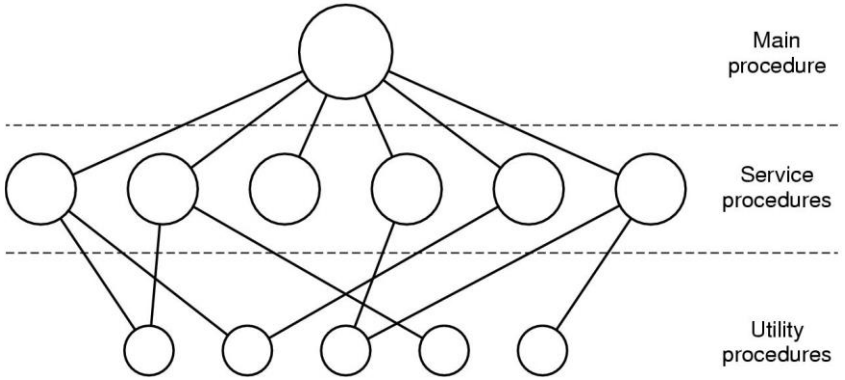
Domanda 6 b2

«**3. Strutture dei Sistemi Operativi**»

Operating Systems: Organizzazione

Monolitico





- Manutenzione difficile
- Vulnerabilità agli errori e agli attacchi
- Blocco del sistema
- Insufficiente protezione fornita dal SO

SO che non hanno una struttura ben definita:

- procedure di servizio compilate in un unico oggetto
- ogni procedura può chiamare tutte le altre
- ogni processo esegue parzialmente in modo kernel
- system call bloccanti

Sapienza Università di Roma

C	Se le funzioni del sistema operativo vengono eseguite all'interno dei processi utente, se un processo effettua una syscall e poi può continuare ad essere eseguito, non avviene alcun process switch
D	Se le funzioni del sistema operativo vengono eseguite all'interno dei processi utente, non c'è bisogno di un process switch per eseguire una funzionalità del sistema operativo

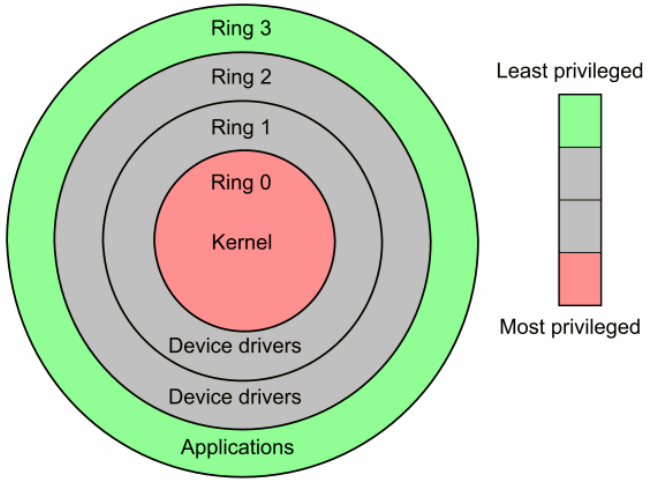
Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

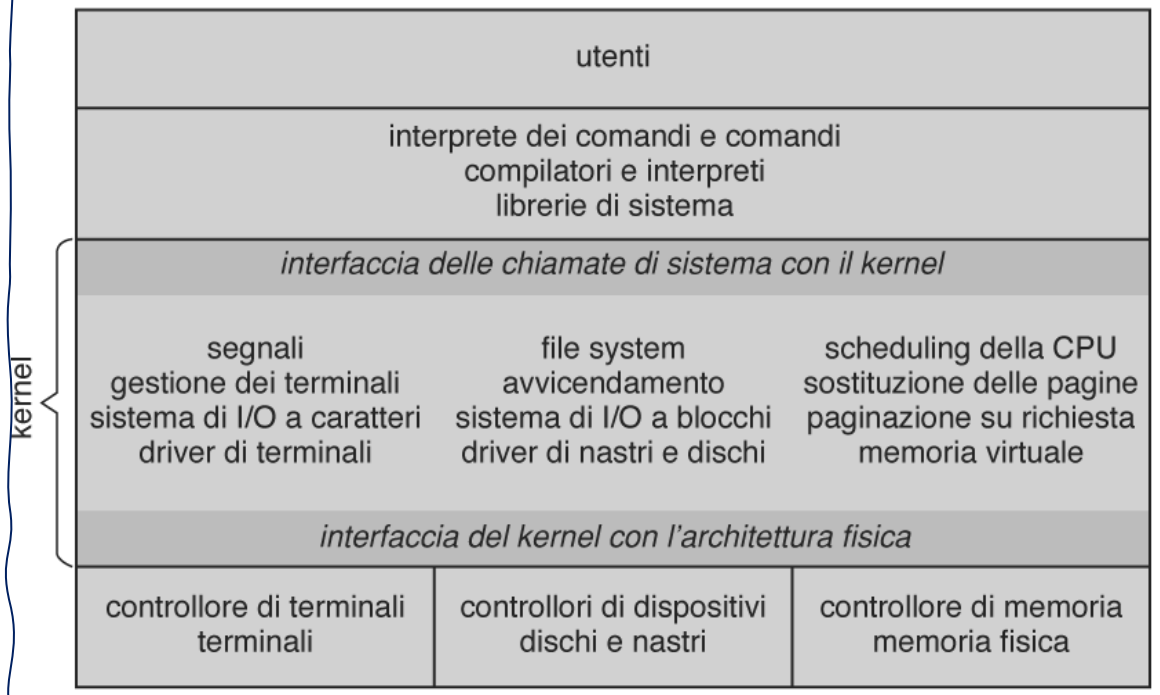
Domanda 6 b3

«**3. Strutture dei Sistemi Operativi**»



Operating Systems: Organizzazione

Stratificato: UNIX



Vantaggi:

- Ogni strato offre una virtualizzazione di un certo numero di funzioni (macchina virtuale)
- La dipendenza dall'hardware è limitata al livello più basso (portabilità)

Svantaggi:

- La portabilità si paga con minor efficienza perché una chiamata di sistema deve attraversare più strati (eventuale adattamento dei dati passati)
- Più difficile da progettare

SAPIENZA UNIVERSITÀ DI ROMA

B	Se le funzioni del sistema operativo vengono eseguite come processi separati, c'è sempre bisogno di un process switch per eseguire una funzionalità del sistema operativo

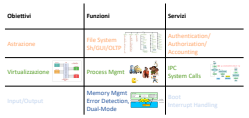
Operating Systems: Esercizi

Domanda 6 b4

«**3. Strutture dei Sistemi Operativi**»

Operating Systems: Organizzazione

MicroKernel



Sposta il maggior numero di funzionalità possibili dal kernel allo spazio utente (come programmi di sistema)

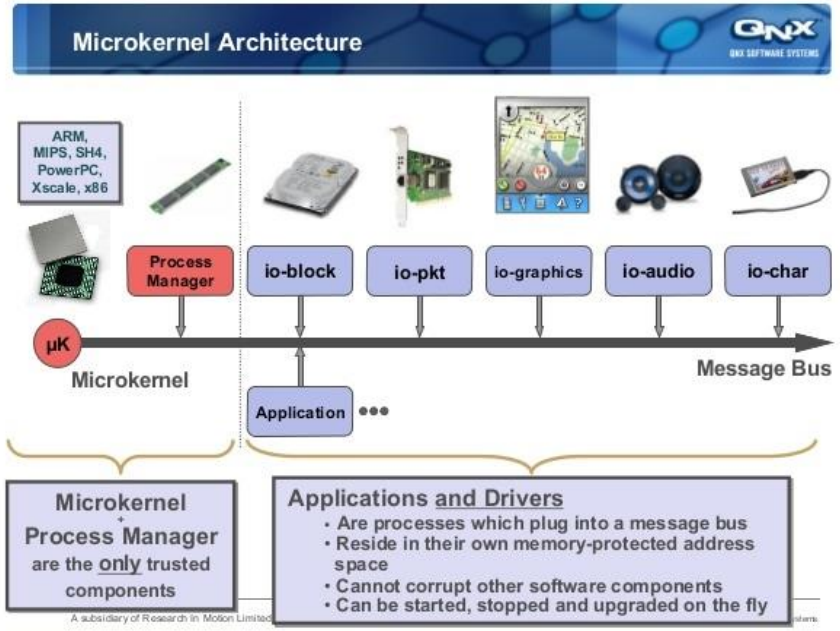
- Il microkernel offre solo servizi per gestire processi, memoria e comunicazione.
- La comunicazione tra moduli avviene tramite scambio di messaggi

Vantaggi:

- Ottimale per i sistemi Real Time (RTOS)
- Facilità di estendere il SO (lasciando invariato il microkernel)
- Maggiore portabilità del SO da un'architettura HW a un'altra
- Maggiore affidabilità e sicurezza (meno codice viene eseguito in modalità kernel)

Svantaggi:

- I microkernel possono soffrire di calo di prestazioni per l'aumento di sovraccarico di funzioni di sistema



SAPIENZA UNIVERSITÀ DI ROMA

AUTOMATICA E GESTIONALE ANTONIO RUBERTI

A	Nel caso di un sistema operativo a kernel separato, la gestione dei process switch è a sua volta un processo

SAPIENZA UNIVERSITÀ DI ROMA

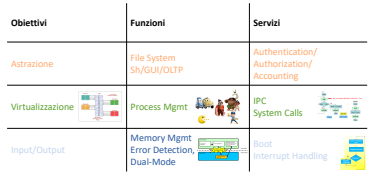
Operating Systems: Esercizi

Domanda 6 b5

«**3. Strutture dei Sistemi Operativi**»

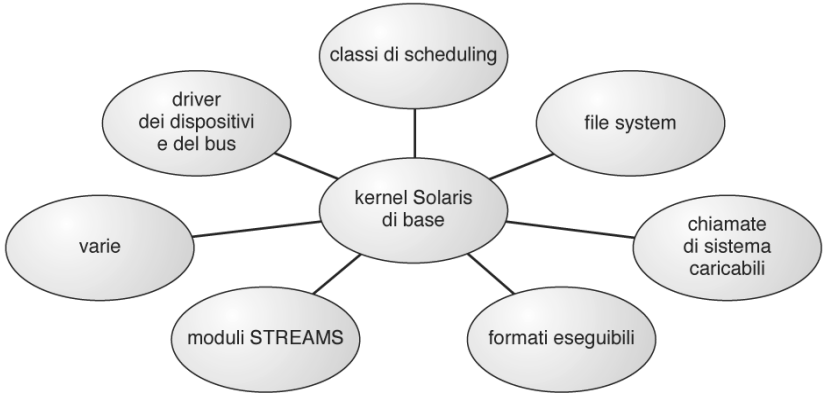
Operating Systems: Organizzazione

Modulare (Kernel a Moduli Funzionali)



Moderna metodologia di progetto che realizza kernel modulari

- attraverso tecniche object-oriented
- ogni componente core è separata e interagisce con le altre attraverso interfacce ben note
- A differenza della struttura a stati i moduli possono comunicare con tutti gli altri, non solo con quelli di livello inferiore
- ogni componente è caricabile dinamicamente nel kernel al momento del boot e/o durante l'esecuzione



Stratificato: maggiormente flessibile
MicroKernel: più efficiente (no messaggi)

SAPIENZA UNIVERSITÀ DI ROMA

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 7

Quale delle seguenti affermazioni sui termini tipici della concorrenza è falsa?

A	Una sezione critica è una porzione di memoria che contiene almeno una variabile condivisa tra più processi
B	Una operazione atomica è una sequenza di istruzioni macchina tale che, se un processo inizia ad eseguirla, allora finirà di eseguirla senza interruzioni da altri processi
C	Il requisito di mutua esclusione prevede che un solo processo possa eseguire un certo segmento di codice o accedere ad una determinata risorsa
D	È possibile che 2 distinti processi chiamino la stessa funzione atomica

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

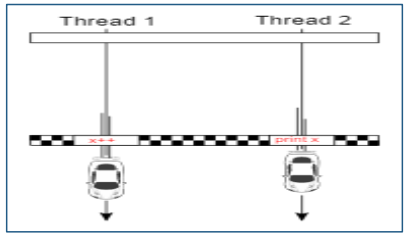
Sapienza Università di Roma

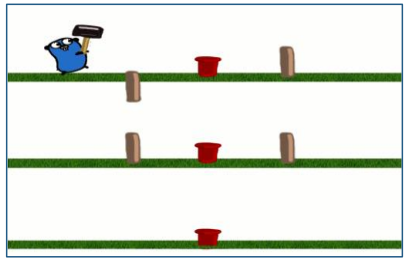
Operating Systems: Esercizi

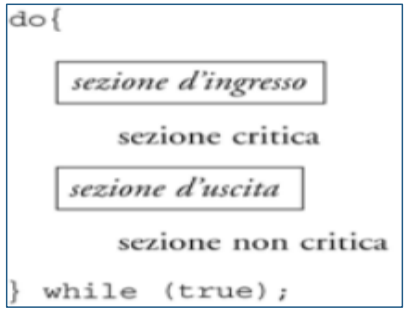
Domanda 7

Operating Systems: IPC

Critical Region: Definizioni







In un Sistema multiprogrammato, multitasking, multithread

- **Race Condition**: (Corsa Critica): situazione in cui il risultato della elaborazione su una risorsa condivisa da più processi, di cui almeno uno vi scrive, dipende da come essi si alternano (run-time: non predicibile, in funzione della schedulazione, degli eventi esterni, degli interrupt, etc).
- **Mutual Exclusion**: (Mutual Esclusione): proibire che più di un processo acceda contemporaneamente alla risorsa condivisa
- **Critical Region** (Sezione Critica): parte del codice di un processo in cui avviene l'accesso alla risorsa condivisa. Per impedire l'ingenerarsi di Race Condition devono essere rispettati i seguenti:
  1. Mutual Exclusion: Un solo processo può essere in una regione critica
  2. Velocità Relative: Le assunzioni su velocità e numero di processori sono ininfluenti
  3. Progresso: Nessun processo la cui elaborazione è al di fuori della sezione critica può fermare altri processi
  4. Bounded Waiting: Nessun processo deve aspettare indefinitamente per entrare nella sua regione critica

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

A	Una sezione critica è una porzione di memoria che contiene almeno una variabile condivisa tra più processi

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 8

Quale delle seguenti affermazioni sugli indirizzi di memoria principale è vera?

A	Un indirizzo fisico fa sempre riferimento alla memoria secondaria
B	Per rispettare il requisito di rilocazione, occorre trasformare indirizzi fisici in logici
C	Gli indirizzi relativi all'inizio dell'immagine di un processo sono usati nella paginazione
D	Nessuna delle altre opzioni è corretta

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 8b

Quale delle seguenti affermazioni sugli indirizzi di memoria principale è vera?

A	Un indirizzo fisico fa sempre riferimento alla memoria secondaria
B	Per rispettare il requisito di rilocazione, occorre trasformare indirizzi fisici in logici
C	Gli indirizzi relativi all'inizio dell'immagine di un processo sono usati nella paginazione
D	Nessuna delle altre opzioni è corretta

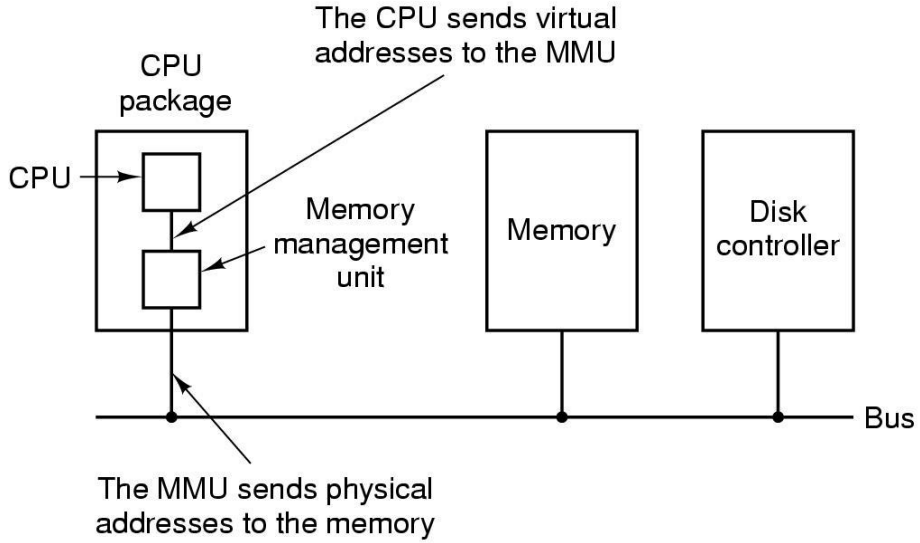
Operating Systems: Memory Management

Virtual Memory: MMU

Memory Manager: mappa gli indirizzi virtuali in indirizzi fisici e li mette sul bus di memoria.

Memory Management Unit

- **Dispositivo hardware** che realizza la trasformazione dagli indirizzi logici a quelli fisici in fase di esecuzione
- •Nello schema di MMU, il valore nel **registro di rilocazione** (r) è aggiunto ad ogni indirizzo logico generato da un processo nel momento in cui è trasmesso alla memoria
- Il **programma** utente interagisce con gli **indirizzi logici** (da 0 a max); non vede mai gli indirizzi fisici reali (da r a r+max)
- In questo modo il programma è indipendente da informazioni specifiche della memoria (valore di rilocazione, capacità della memoria, …)



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 9

Quale delle seguenti affermazioni sulla memoria virtuale con paginazione è falsa?

A	Diminuire la dimensione delle pagine ha effetti positivi sul numero di pagine che possono trovarsi in memoria principale
B	Aumentare la dimensione delle pagine ha effetti positivi sulla frammentazione interna
C	Diminuire la dimensione delle pagine ha effetti negativi sulla dimensione della tabella delle pagine
D	Aumentare la dimensione delle pagine ha effetti negativi sulla multiprogrammazione

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 9b1

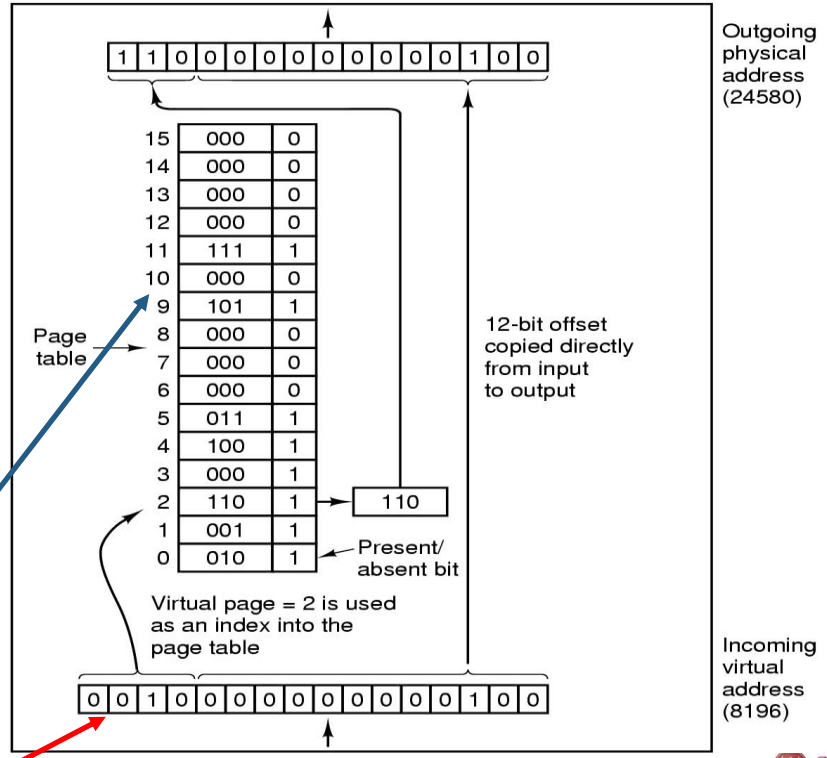
Operating Systems: Memory Management

Virtual Memory: Paging 1/7

Memory Manager: suddivide la memoria in porzioni uguali e facilmente usabili.

Paginazione

- **Efficienza**: I meccanismi a partizionamento fisso/dinamico non sono efficienti nell'uso della memoria (frammentazione interna/esterna)
- **Riduzione della Frammentazione**: allocando ai processi spazio di memoria non contiguo
- **Aumento del Degree of Multiprogramming**: si tiene in memoria solo una porzione del programma, aumentando il numero dei processi che possono essere contemporaneamente presenti in memoria
- **Memoria Virtuale**: possibilità di eseguire un processo più grande della memoria disponibile
- **MMU**: pieno utilizzo del dispositivo HW



A	Diminuire la dimensione delle pagine ha effetti positivi sul numero di pagine che possono trovarsi in memoria principale
C	Diminuire la dimensione delle pagine ha effetti negativi sulla dimensione della tabella delle pagine

Operating Systems: Esercizi

Domanda 9b2

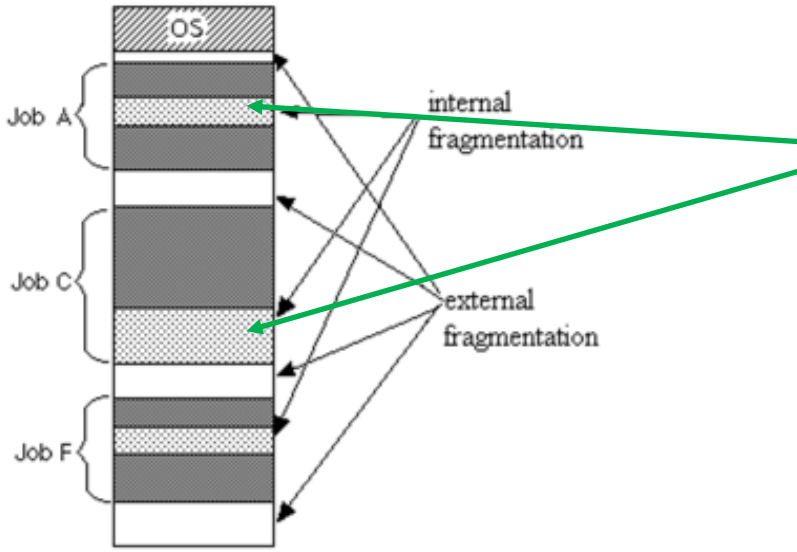
Operating Systems: Memory Management

Basic: Memory Partitioning 4/4

Memory Manager: gestione delle porzioni di memoria frammentate (che non riescono ad essere utilizzate).

Frammentazione

- **Interna**: lo spazio allocato in eccesso rispetto alle esigenze dei processi, inutilizzabile perché allocato. Inevitabile. Se ne può ridurre l'impatto riducendo la dimensione delle partizioni
- **Esterna**: lo spazio libero in aree troppo piccole per essere utili. Può essere ridotta tramite:
  - **Rilocazione**: tecniche di compattamento. Efficaci ma computazionalmente pesanti
  - **Paginazione**: partizioni di dimensioni fisse



Aumentare la dimensione delle pagine ha effetti **negativi** sulla frammentazione interna

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

B	Aumentare la dimensione delle pagine ha effetti positivi sulla frammentazione interna

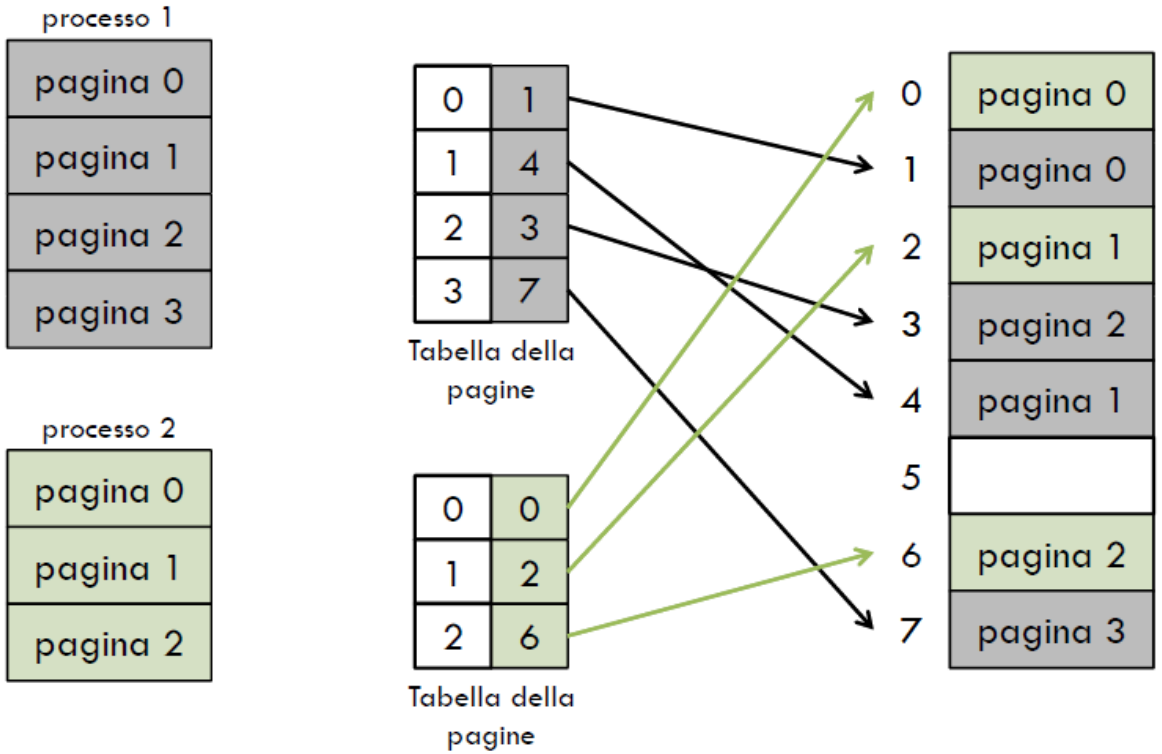
Operating Systems: Esercizi

Domanda 9b3

Operating Systems: Memory Management

Virtual Memory: Paging 5/7

Memory Manager: suddivide la memoria in porzioni uguali e facilmente usabili.



Esempio

La tabella delle pagine associa l'indirizzo di una pagina logica al corrispettivo indirizzo della pagina nella memoria fisica

Le aree di memoria dei processi possono essere interfogliate

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

D	Aumentare la dimensione delle pagine ha effetti negativi sulla multiprogrammazione

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 9d

Quale delle seguenti affermazioni sulla memoria virtuale con paginazione è **falsa**?

A	~~Diminuire la dimensione delle pagine ha effetti positivi sul numero di pagine che possono trovarsi in memoria principale~~		A parità di dimensione della memoria, pagine più piccole sono in numero maggiore
B	Aumentare la dimensione delle pagine ha effetti positivi sulla frammentazione interna		Aumentare la dimensione delle pagine ha effetti negativi sulla frammentazione interna
C	~~Diminuire la dimensione delle pagine ha effetti negativi sulla dimensione della tabella delle pagine~~		A parità di dimensione della memoria, pagine più piccole sono in numero maggiore: richiedono tabella più grande
D	~~Aumentare la dimensione delle pagine ha effetti negativi sulla multiprogrammazione~~		A parità di dimensione della memoria, pagine più piccole sono in numero minore: meno processi in memoria

WHY

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 10

Quale delle seguenti affermazioni sulla traduzione di un indirizzo virtuale in fisico, in un sistema con memoria virtuale con paginazione (avente tabella delle pagine ad 1 livello), è falsa?

A	L'hardware deve anche estrarre dall'indirizzo virtuale il numero di pagina virtuale; tale operazione è equivalente ad una divisione intera
B	L'hardware deve anche usare il numero di pagina per accedere alla tabella delle pagine del processo in esecuzione. A tal proposito, deve conoscere l'inizio di tale tabella, che viene definito dal software (sistema operativo). Tale indirizzo può cambiare durante l'esecuzione del processo: sta al sistema operativo mantenerlo aggiornato
C	L'hardware deve anche usare il numero di frame ottenuto dalla tabella delle pagine per comporre, insieme con l'offset originale, l'indirizzo fisico. Tale operazione è equivalente ad uno shift seguito da una somma
D	L'hardware deve anche cercare il numero di pagina nelle entries della tabella delle pagine del processo in esecuzione.

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 10b1

Per non effettuare una divisione, il numero delle pagine deve essere una potenza di 2 (➔ shift).

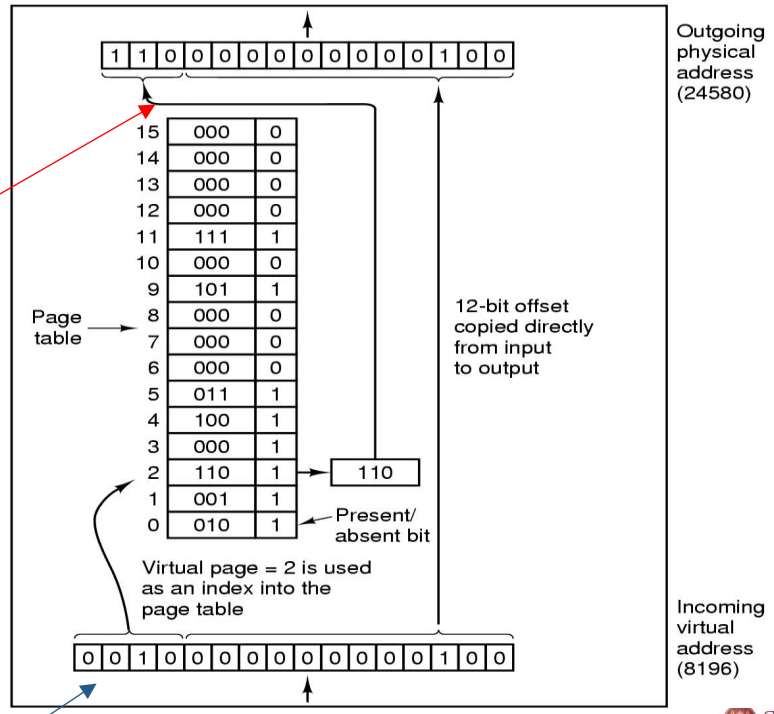
Operating Systems: Memory Management

Virtual Memory: Paging 1/7

Memory Manager: suddivide la memoria in porzioni uguali e facilmente usabili.

Paginazione

- **Efficienza**: I meccanismi a partizionamento fisso/dinamico non sono efficienti nell'uso della memoria (frammentazione interna/esterna)
- **Riduzione della Frammentazione**: allocando ai processi spazio di memoria non contiguo
- **Aumento del Degree of Multiprogramming**: si tiene in memoria solo una porzione del programma, aumentando il numero dei processi che possono essere contemporaneamente presenti in memoria
- **Memoria Virtuale**: possibilità di eseguire un processo più grande della memoria disponibile
- **MMU**: pieno utilizzo del dispositivo HW



A	L'hardware deve anche estrarre dall'indirizzo virtuale il numero di pagina virtuale; tale operazione è equivalente ad una divisione intera
C	L'hardware deve anche usare il numero di frame ottenuto dalla tabella delle pagine per comporre, insieme con l'offset originale, l'indirizzo fisico. Tale operazione è equivalente ad uno shift seguito da una somma

Operating Systems: Esercizi

Domanda 10b2

B	L'hardware deve anche usare il numero di pagina per accedere alla tabella delle pagine del processo in esecuzione. A tal proposito, deve conoscere l'inizio di tale tabella, che viene definito dal software (sistema operativo). Tale indirizzo può cambiare durante l'esecuzione del processo: sta al sistema operativo mantenerlo aggiornato

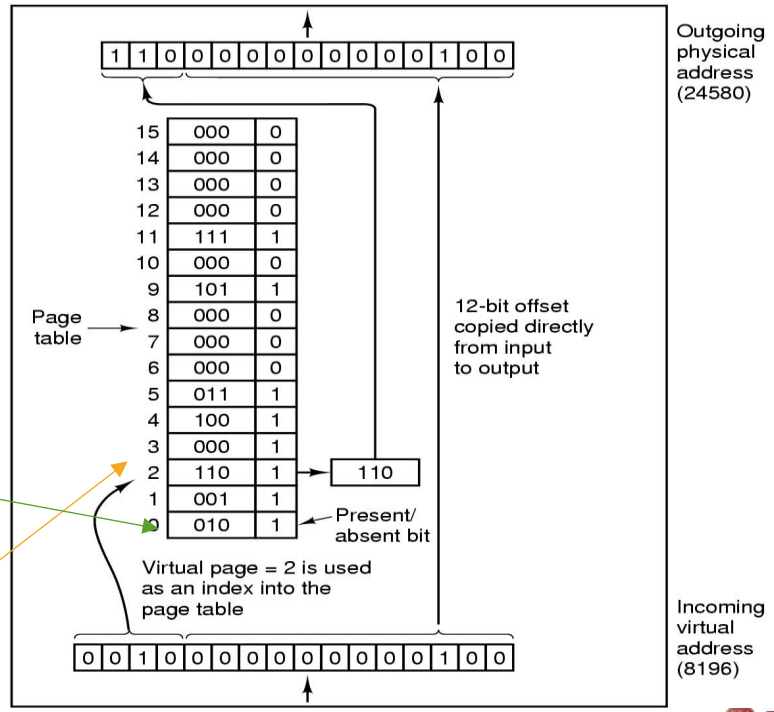
Operating Systems: Memory Management

Virtual Memory: Paging 1/7

Memory Manager: suddivide la memoria in porzioni uguali e facilmente usabili.

Paginazione

- **Efficienza**: I meccanismi a partizionamento fisso/dinamico non sono efficienti nell'uso della memoria (frammentazione interna/esterna)
- **Riduzione della Frammentazione**: allocando ai processi spazio di memoria non contiguo
- **Aumento del Degree of Multiprogramming**: si tiene in memoria solo una porzione del programma, aumentando il numero dei processi che possono essere contemporaneamente presenti in memoria
- **Memoria Virtuale**: possibilità di eseguire un processo più grande della memoria disponibile
- **MMU**: pieno utilizzo del dispositivo HW



D	L'hardware deve anche cercare il numero di pagina nelle entries della tabella delle pagine del processo in esecuzione

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 10b3

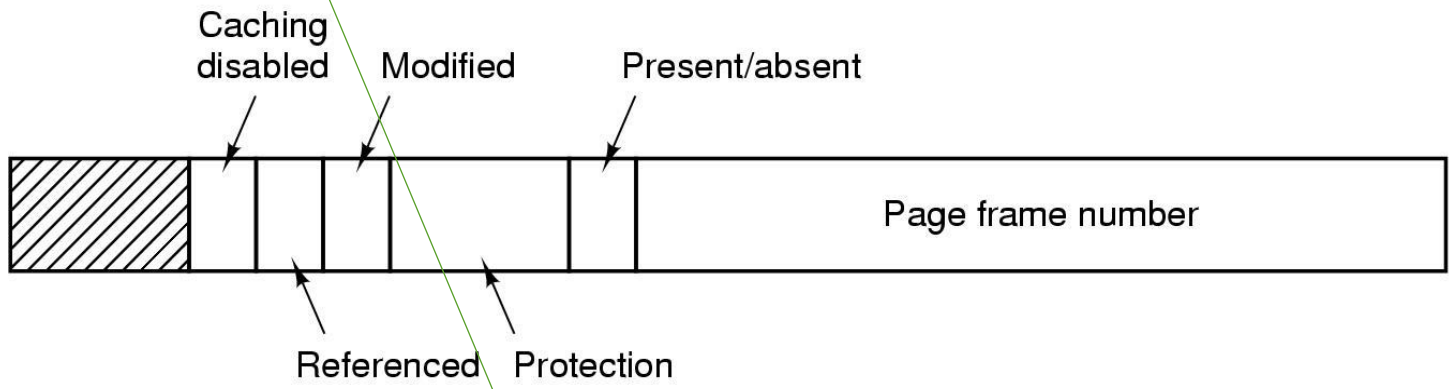
Operating Systems: Memory Management

Virtual Memory: Paging 4/7

Memory Manager: suddivide la memoria in porzioni uguali e facilmente usabili.

Tabella delle Pagine (referenziata nel PCB)

Oltre a Page (p) e Displacement (d), per ogni pagina referenziata contiene le informazioni



- **Present/Absent**: attualmente contenuta in memoria oppure no (➔ necessario accesso a disco)
- **Protection**: modalità di protezione (es. r, w, x)
- **Modified**: modificata di recente (deve essere salvata su disco) oppure no. Detto anche «**Dirty Bit**»
- **Referenced**: è stata letta di recente (probabilmente lo sarà ancora: non conviene segregarla su disco)
- **Caching Disabled**: la copia su memoria è aggiornata rispetto alla quella in cache

SAPIENZA UNIVERSITÀ DI ROMA

B	L'hardware deve anche usare il numero di pagina per accedere alla tabella delle pagine del processo in esecuzione. A tal proposito, deve conoscere l'inizio di tale tabella, che viene definito dal software (sistema operativo). Tale indirizzo può cambiare durante l'esecuzione del processo: sta al sistema operativo mantenerlo aggiornato

Operating Systems: Esercizi

Domanda 10c

Quale delle seguenti affermazioni sulla traduzione di un indirizzo virtuale in fisico, in un sistema con memoria virtuale con paginazione (avente tabella delle pagine ad 1 livello), è **falsa**?

A	L'hardware deve anche estrarre dall'indirizzo virtuale il numero di pagina virtuale; tale operazione è equivalente ad una divisione intera
B	~~L'hardware deve anche usare il numero di pagina per accedere alla tabella delle pagine del processo in esecuzione. A tal proposito, deve conoscere l'inizio di tale tabella, che viene definito dal software (sistema operativo). Tale indirizzo può cambiare durante l'esecuzione del processo: sta al sistema operativo mantenerlo aggiornato~~
C	~~L'hardware deve anche usare il numero di frame ottenuto dalla tabella delle pagine per comporre, insieme con l'offset originale, l'indirizzo fisico. Tale operazione è equivalente ad uno shift seguito da una somma~~
D	~~L'hardware deve anche cercare il numero di pagina nelle entries della tabella delle pagine del processo in esecuzione.~~

A	Shift seguito da somma (potenza di 2)
B	OK
C	OK
D	OK

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 11

Quale delle seguenti affermazioni sulla concorrenza tra processi o thread è falsa?

A	La disabilitazione delle interruzioni impedisce la creazione di nuove interruzioni
B	Se un processo utente può disabilitare le interruzioni tramite un'istruzione macchina dedicata, allora può far diminuire l'uso del processore
C	La disabilitazione delle interruzioni non funziona su sistemi con più processori o più core
D	L'abuso della disabilitazione delle interruzioni fa diminuire la multiprogrammazione, a parità di numero di processi

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 11b

Operating Systems: IPC

Mutual Exclusion with Busy Waiting 1/3

- **Interrupt Disabling**: (Disabilitazione delle Interruzioni): il processo fa una esplicita richiesta di disabilitazione di tutte le interruzioni, durante la esecuzione della sezione critica, in modo da non incorrere in inconsistenza di dati sulla risorsa condivisa.
  - ➔ Operazione effettuabile solo dal kernel (pericoloso se eseguito da un processo-utente);
  - ➔ Non funziona sui multi-core perché la disabilitazione degli interrupt ha effetto solo su una CPU alla volta

- **Lock Variable**: (Variabile di Blocco): stato inerente la esecuzione di sezioni critiche. Variabile condivisa che definisce lo stato (0: risorsa libera; 1: risorsa in uso) di uso di una risorsa: segnala se è in uso da parte di un processo (nella sua sezione critica).
  - ➔ **Busy Waiting**: i processi al di fuori della sezione critica continuano a testare la variabile fino a che lock non ridiventi 0;
  - ➔ **Strict Alternation**: l'utilizzo di una sola variabile permette di descrivere solo 2 stati (processo A in Critical Region – processo B in Critical Region). Esiste anche l'evenienza che nessuno sia in critical region;

```
do{
    acquisisce il lock
        sezione critica
    restituisce il lock
        sezione non critica
} while (true);
```

SAPIENZA UNIVERSITÀ DI ROMA

A	La disabilitazione delle interruzioni impedisce la creazione di nuove interruzioni
B	Se un processo utente può disabilitare le interruzioni tramite un'istruzione macchina dedicata, allora può far diminuire l'uso del processore
C	La disabilitazione delle interruzioni non funziona su sistemi con più processori o più core

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

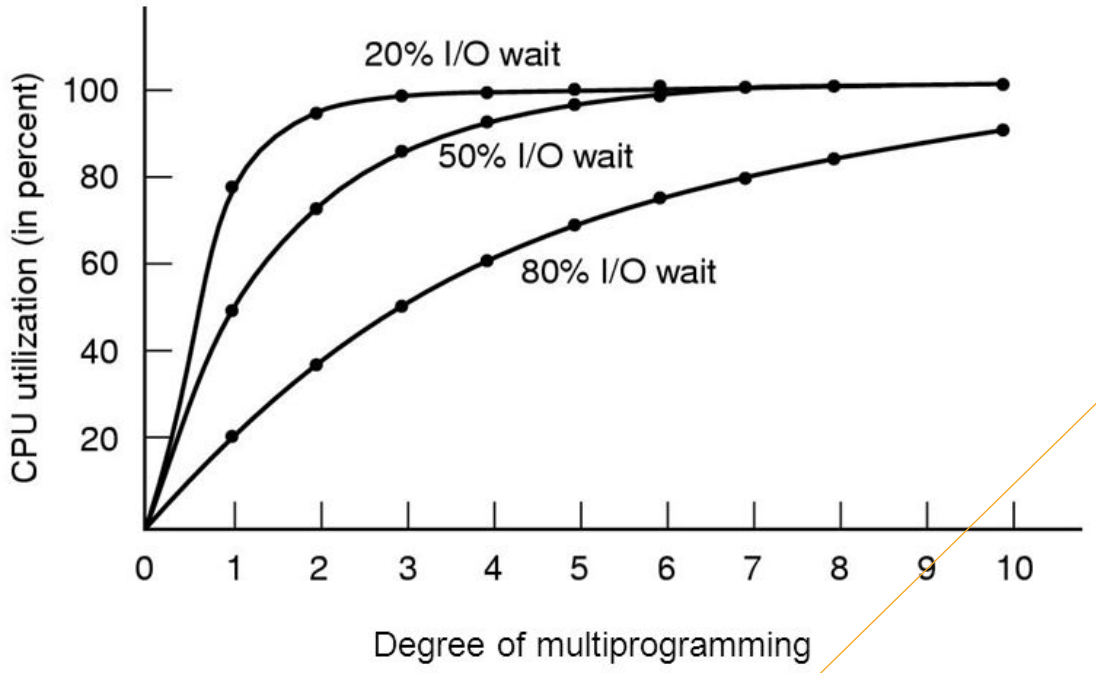
Domanda 11c

Grado di multiprogrammazione: funzione del numero di processi (rimane invariata)

Operating Systems: Memory Management

Basic: Degree of Multiprogramming 3/4

Memory Manager: tenere presente anche il livello di utilizzazione della CPU.



Assumendo che:

- Tutti gli *n* processi in esecuzione presentano la stessa percentule *p* in stato «Wait or Block»
- La condizione di «Wait or Block» è fondamentalmente causata da **I/O Wait**

Si definisce **Degree of Multiprogramming** la funzione di n:

$CPU\text{-}utilization = 1 - p^n$

p	*n (40%)*	*n (90%)*
20%	1	2
50%	2	5
80%	3	10

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

D	L'abuso della disabilitazione delle interruzioni fa diminuire la multiprogrammazione, a parità di numero di processi

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 12

Quale delle seguenti affermazioni, riguardanti la classificazione delle risorse di un sistema operativo e la loro relazione con il deadlock, è vera?

A	Nel caso delle risorse riusabili, se c'è un deadlock allora è stata richiesta almeno una risorsa non ancora creata
B	Nel caso delle risorse riusabili, se c'è un deadlock allora c'è una successione circolare di processi, ciascuno dei quali richiede una risorsa al processo successivo, che però la deve ancora creare
C	Nel caso delle risorse consumabili, se c'è un deadlock allora c'è una successione circolare di processi, ciascuno dei quali richiede una risorsa al processo successivo, che però la deve ancora creare
D	Nel caso delle risorse consumabili, se c'è un deadlock allora è stata richiesta almeno una risorsa già detenuta da un altro processo

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 12: Deadlock: risorse riusabili e consumabili (“usa e getta”)

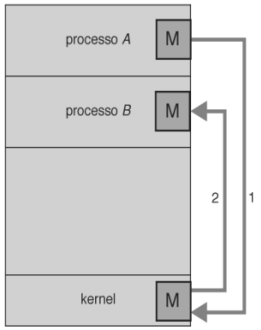
In «**5. Thread e IPC**» gli IPC forniscono un esempio delle due

Operating Systems: IPC

IPC: Implementazioni a Scambio di Messaggi

- Quantità di informazioni da trasmettere
- Velocità di esecuzione
- Scalabilità
- Semplicità di uso nelle applicazioni
- Omogeneità delle comunicazioni
- Integrazione nel linguaggio di programmazione
- Affidabilità
- Sicurezza
- Protezione

(es. Segnale, Socket, Message Queue, Semaforo)



SAPIENZA UNIVERSITÀ DI ROMA

Risorse Consumabili

(usa e getta)

Create (es. scrittura del messaggio)

Consumate (es. lettura del messaggio)

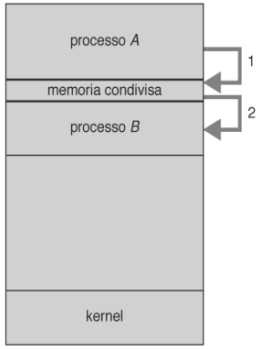
➔ Le informazioni sono presenti solo se create e non consumate

Operating Systems: IPC

IPC: Implementazioni a Memoria Condivisa

- Quantità di informazioni da trasmettere
- Velocità di esecuzione
- Scalabilità
- Semplicità di uso nelle applicazioni
- Omogeneità delle comunicazioni
- Integrazione nel linguaggio di programmazione
- Affidabilità
- Sicurezza
- Protezione

(es. Pipe, Memoria Condivisa)



SAPIENZA UNIVERSITÀ DI ROMA

Risorse Riusabili

Le informazioni sono sempre ivi presenti

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 12: Deadlock: risorse riusabili

In «**03-SOReCa. Deadlock**» si fa riferimento solo alle risorse riusabili (es. CPU, I/O, semaphore).

Per le risorse consumabili, vale lo stesso ragionamento, considerando che la mancata creazione ➔ blocco.

Inoltre, **non** ha senso parlare di:

- **creazione/distruzione** di risorse riusabili (le informazioni sono già ivi presenti).
- **detenzione** di risorse consumabili

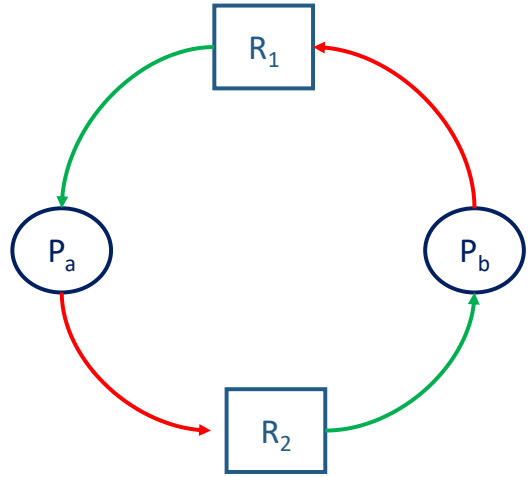
Operating Systems: Deadlock

Deadlock (Stallo): Definizione

➔ **Stallo**: un gruppo di processi entra in uno stallo quando tutti i processi del gruppo attendono il rilascio di una risorsa che può essere liberata solo da uno dei processi in attesa. Esempio con due processi (P_a e P_b) e due risorse (R_1 e R_2)

- Il processo P_a ottiene il possesso della risorsa R_1
- Il processo P_b ottiene il possesso della risorsa R_2
- Il processo P_a richiede la risorsa R_2

➔ Il processo P_b richiede la risorsa R_1



SAPIENZA UNIVERSITÀ DI ROMA

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 12d

Quale delle seguenti affermazioni, riguardanti la classificazione delle risorse di un sistema operativo e la loro relazione con il deadlock, è vera?

A	Nel caso delle risorse riusabili, se c'è un deadlock allora è stata richiesta almeno una risorsa non ancora creata	**KO**	Riusabile + creata!
B	Nel caso delle risorse riusabili, se c'è un deadlock allora c'è una successione circolare di processi, ciascuno dei quali richiede una risorsa al processo successivo, che però la deve ancora creare	**KO**	Riusabile + creare!
C	Nel caso delle risorse consumabili, se c'è un deadlock allora c'è una successione circolare di processi, ciascuno dei quali richiede una risorsa al processo successivo, che però la deve ancora creare		Nel caso delle risorse consumabili, se c'è un deadlock allora c'è una successione circolare di processi, ciascuno dei quali richiede una risorsa al processo successivo, che però la deve ancora creare
D	Nel caso delle risorse consumabili, se c'è un deadlock allora è stata richiesta almeno una risorsa già detenuta da un altro processo	**KO**	Consumabile + detenere!

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 13

Assumendo un sistema monoprocessore, quale delle seguenti affermazioni sulla preemption è vera?

A	Se uno scheduler è non-preemptive, è possibile che un processo monopolizzi il processore, anche in presenza di altri processi ready
B	Se uno scheduler è non-preemptive, permette sempre ai suoi processi di essere eseguiti senza interruzioni sul processore fino al loro completamento
C	Se uno scheduler è preemptive e vi è più di 1 processo ready, non è possibile che un processo monopolizzi il processore
D	Per avere un trattamento equo sui processi, è sufficiente usare uno scheduler preemptive

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 13b1

In «**05-SOReCa. Scheduling**»

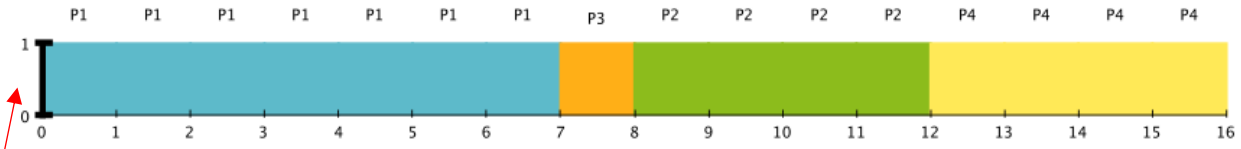
Operating Systems: Scheduling

Batch: Shortest Remaining Time First

due schemi:

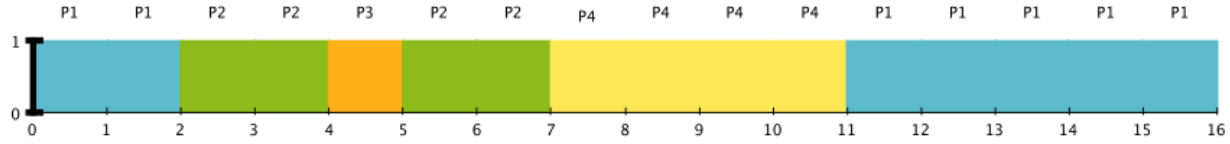
- Non-preemptive (**SJF**): quando un processo arriva nella coda dei processi pronti mentre il processo precedente è ancora in esecuzione, l'algoritmo permette al processo corrente di finire il suo uso della CPU

Tempo di attesa medio =
[0 + (8-2) + (7-4) + (12-5)]/4 = 4



- Preemptive (**SRTF**): quando un processo arriva nella coda dei processi pronti con un tempo di computazione minore del tempo che rimane al processo correntemente in esecuzione, l'algoritmo ferma il processo corrente. Questa schedulazione è anche detta shortest-remaining-time-first

Tempo di attesa medio =
[(11-2) + 1 + 0 +(7-5))/4 = 3



SAPIENZA UNIVERSITÀ DI ROMA

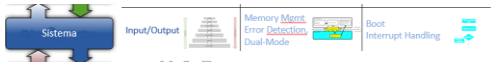
A Se uno scheduler è non-preemptive, è possibile che un processo monopolizzi il processore, anche in presenza di altri processi ready

Operating Systems: Esercizi

Domanda 13b

Operating Systems: Input/Output

I/O Software: Interrupt



Interrupt Driven I/O: ogni I/O necessita l'interruzione della elaborazione corrente

In «**07-SOReCa-StorageFSO**»

Interrupt: interrompe l'esecuzione della elaborazione

```
copy_from_user(buffer, p, count);
enable_interrupts( );
while (*printer_status_reg != READY) ;
*printer_data_register = p[0];
scheduler( );
```

(a)

```
if (count == 0) {
    unblock_user( );
} else {
    *printer_data_register = p[i];
    count = count – 1;
     i = i + 1;
}
acknowledge_interrupt( );
return_from_interrupt( );
```

(b)

Scrivere una stringa alla stampante usando l'I/O guidato da interrupt.
(a) Codice eseguito al momento della chiamata al sistema di stampa.
(b) Procedura di servizio d'interruzione per la stampante

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

B	Se uno scheduler è non-preemptive, permette sempre ai suoi processi di essere eseguiti senza interruzioni sul processore fino al loro completamento

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 13b

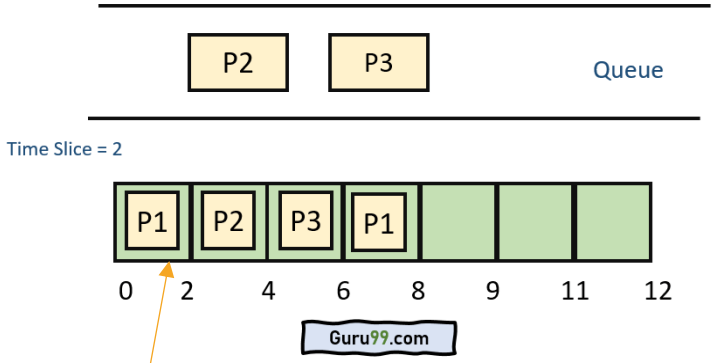
In «**05-SOReCa. Scheduling**»

Operating Systems: Scheduling

Algoritmi per sistemi Interattivi

- Round-robin scheduling
- Priority scheduling
- Multiple queues
- Shortest process next
- Guaranteed scheduling

Usually, PreEmptive



Quanti di tempo per l'allocazione della CPU

C	Se uno scheduler è preemptive e vi è più di 1 processo ready, non è possibile che un processo monopolizzi il processore
D	Per avere un trattamento equo sui processi, è sufficiente usare uno scheduler preemptive

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

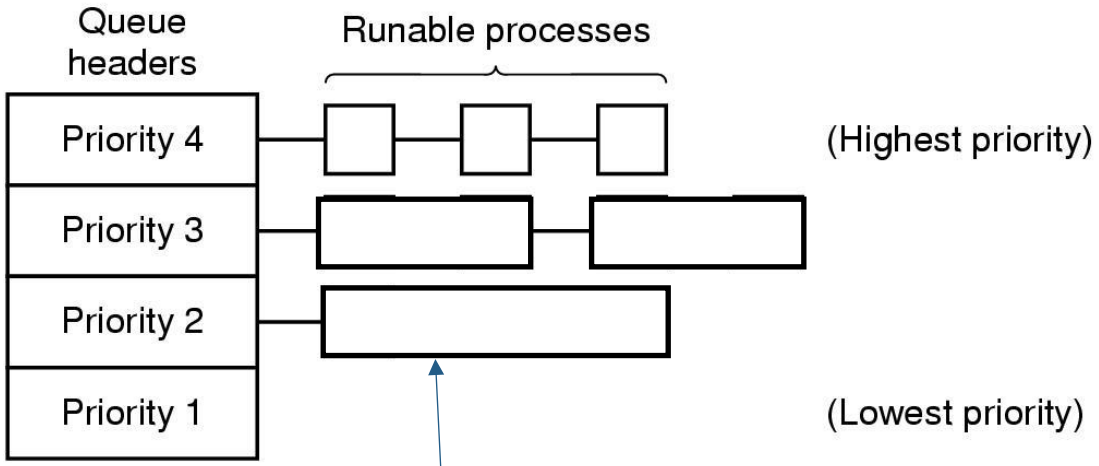
Domanda 13b

In «**05-SOReCa. Scheduling**»

Operating Systems: Scheduling

Interattivi: Multiple queue

Analogo a «Priority Scheduling» ma i lavori più brevi hanno una priorità maggiore e quanti di tempo minori.



Il comando Unix "nice" permette all'utente di abbassare la priorità del lavoro volontariamente.
Di conseguenza, i lavori più brevi (ad alta priorità) escono dalla CPU per primi
Si annunciano da soli - non si presuppone una conoscenza precedente!.

SAPIENZA UNIVERSITÀ DI ROMA

C	Se uno scheduler è preemptive e vi è più di 1 processo ready, non è possibile che un processo monopolizzi il processore
D	Per avere un trattamento equo sui processi, è sufficiente usare uno scheduler preemptive

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 14

Quale delle seguenti affermazioni sui meccanismi per la gestione della concorrenza è vera?

A	Usando i semafori di qualsiasi tipo, è possibile scrivere processi che non soffrano di starvation
B	Disabilitando gli interrupt, è possibile scrivere processi che non soffrano di starvation
C	Senza usare né semafori, né scambio messaggi, né istruzioni macchina atomiche, è possibile scrivere processi che non soffrano di starvation per garantire la mutua esclusione tra 2 process
D	Usando le istruzioni macchina exchange e compare_and_swap, è possibile scrivere processi che non soffrano di starvation

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

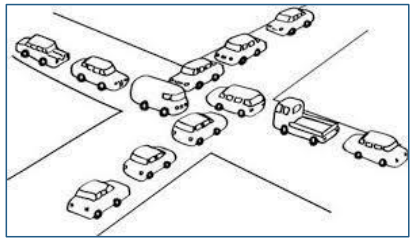
Operating Systems: Esercizi

Domanda 14b1

Definizioni

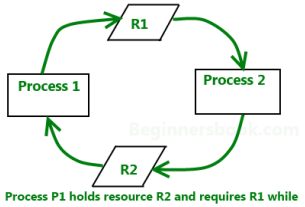
Operating Systems: IPC

Critical Region: Definizioni



In un Sistema multiprogrammato, multitasking, multithread

- **Deadlock**: (Stallo): situazione in cui due o più processi non possono procedere con la prossima istruzione, perché ciascuno attende l'altro.



- **LiveLock**: (Stallo Attivo): situazione in cui due o più processi cambiano continuamente il proprio stato, l'uno in risposta all'altro, senza fare alcunché di "utile"

- **Starvation** (Inedia): un processo, pur essendo ready, non viene mai scelto dallo scheduler

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 14b2

Disabilitando gli interrupt (pratica non educata), si evita l'inconsistenza di dati

Operating Systems: IPC

Mutual Exclusion with Busy Waiting 1/3

- **Interrupt Disabling**: (Disabilitazione delle Interruzioni): il processo fa una esplicita richiesta di disabilitazione di tutte le interruzioni, durante la esecuzione della sezione critica, in modo da non incorrere in inconsistenza di dati sulla risorsa condivisa.
  - ➔ Operazione effettuabile solo dal kernel (pericoloso se eseguito da un processo-utente);
  - ➔ Non funziona sui multi-core perché la disabilitazione degli interrupt ha effetto solo su una CPU alla volta

- **Lock Variable**: (Variabile di Blocco): stato inerente la esecuzione di sezioni critiche. Variabile condivisa che definisce lo stato (0: risorsa libera; 1: risorsa in uso) di uso di una risorsa: segnala se è in uso da parte di un processo (nella sua sezione critica).
  - ➔ **Busy Waiting**: i processi al di fuori della sezione critica continuano a testare la variabile fino a che lock non ridiventi 0;
  - ➔ **Strict Alternation**: l'utilizzo di una sola variabile permette di descrivere solo 2 stati (processo A in Critical Region – processo B in Critical Region). Esiste anche l'evenienza che nessuno sia in critical region;

```
do{
    acquisisce il lock
        sezione critica
    restituisce il lock
        sezione non critica
} while (true);
```

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

B	Disabilitando gli interrupt, è possibile scrivere processi che non soffrano di starvation

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

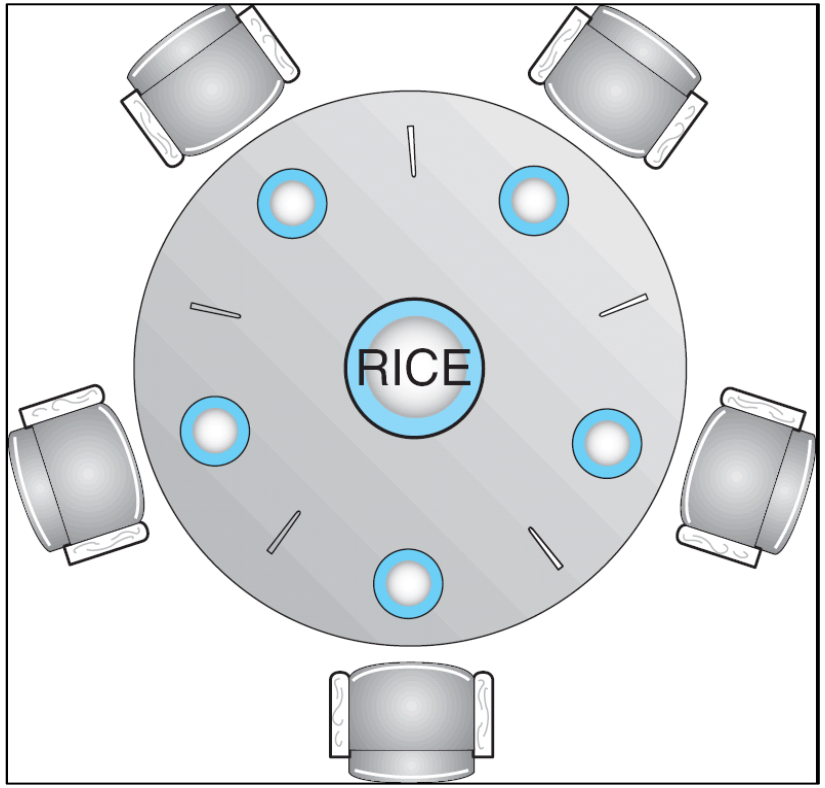
Domanda 14b3

Il problema dei «5 filosofi a cena» esemplifica alcuni concetti inerenti la gestione della concorrenza. In particolare:

- Deadlock
- Starvation

Operating Systems: IPC

Dining Philosopher Problem 1/5



I 5 filosofi a cena:

- 5 filosofi su 5 sedie
- 5 bacchette per prendere il riso posto in una zuppiera condivisa
- Un filosofo può prendere una bacchetta alla volta, solo se è libera e se è posta tra lui e un vicino
- Quando il filosofo non mangia pensa e non interagisce con gli altri
- Quando il filosofo ha due bacchette mangia, dopo di che lascia le bacchette e torna a pensare

➔ Deve essere possibile che 2 filosofi possano mangiare contemporaneamente

➔ Processi in competizione per l'accesso esclusivo a risorse in numero limitato

Operating Systems: Esercizi

Domanda 14b4

Operating Systems: IPC

Dining Philosopher Problem 3/5


```
#define N 5                 /*number of philosophers*/
typedef int semaphore;
semaphore mutex = 1;
void philosopher(int i)      /*i:philosopher number, from 0 to 4*/
{
while(TRUE){
        think();       /*philosopher is thinking*/
        down (&mutex) ;       /* entering critical section */
        take_fork(i); /*take left fork*/
        take_fork(i+1)%N;    /*take right for;% is modulo operator*/
        eat():         /*self-explanatory*/
        put_fork(i);  /*put left fork back on table*/
        put_fork(i+1)%N;     /*put right fork back on table*/
        up (&mutex)    /* exiting critical section */
        }
}
```

Soluzione non efficiente: solo 1 filosofo per volta mangia, non 2!

SAPIENZA UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

A	Usando i semafori di qualsiasi tipo, è possibile scrivere processi che non soffrano di starvation

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 14b5

Senza semafori ➔

- Starvation
- Deadlock

… per più processi.

Operating Systems: IPC

Dining Philosopher Problem 2/5


```
#define N 5                 /*number of philosophers*/
Void philosopher(int i)     /*i:philosopher number, from 0 to 4*/
{
While(TRUE){
        think();        /*philosopher is thinking*/
        take_fork(i); /*take left fork*/
        take_fork(i+1)%N;    /*take right for;% is modulo operator*/
        eat():          /*self-explanatory*/
        put_fork(i);  /*put left fork back on table*/
        put_fork(i+1)%N;     /*put right fork back on table*/
        }
}
```

Non soluzione:

- **starvation**: se tutti i filosofi prendono la forchetta sinistra, nessuno riesce a prendere la destra, nessuno mangia
- **deadlock**: tutti i filosofi sono in attesa di una azione da parte degli altri

SAPIENZA UNIVERSITÀ DI ROMA

C	Senza usare né semafori, né scambio messaggi, né istruzioni macchina atomiche, è possibile scrivere processi che non soffrano di starvation per garantire la mutua esclusione tra 2 process

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI
SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Esercizi

Domanda 14b6

… ma nel caso di solo 2 processi
➔ **Algoritmo di Peterson**

Operating Systems: IPC

Mutual Exclusion with Busy Waiting 2/3

- **Turn Variable** (Variabile di Turno): definisce il turno di uso della risorsa. Annovera anche la situazione in cui nessun processo è interessato alla sezione critica. Da usarsi assieme alla variabile lock. Inizialmente sviluppato da Dekker e documentato da Dijkstra nel 1965, è stato ulteriormente ottimizzato da Peterson nel 1981.
  - ➔ **Busy Waiting**: i processi interessati ma alla sezione critica ma al di fuori di essa continuano a testare la variabile fino a che lock non ridiventi 0;

```
#define FALSE  0
#define TRUE   1
#define N      2                        /* number of processes */

int turn;                               /* whose turn is it? */
int interested[N];                      /* all values initially 0 (FALSE) */

void enter_region(int process);         /* process is 0 or 1 */
{
     int other;                         /* number of the other process */

     other = 1 – process;               /* the opposite of process */
     interested[process] = TRUE;        /* show that you are interested */
     turn = process;                    /* set flag */
     while (turn == process && interested[other] == TRUE)  /* null statement */ ;
}

void leave_region(int process)          /* process: who is leaving */
{
     interested[process] = FALSE;       /* indicate departure from critical region */
}
```

Figure 2-24. Peterson's solution for achieving mutual exclusion.

SAPIENZA UNIVERSITÀ DI ROMA

C	Senza usare né semafori, né scambio messaggi, né istruzioni macchina atomiche, è possibile scrivere processi che non soffrano di starvation per garantire la mutua esclusione tra 2 process

Operating Systems: Esercizi

Domanda 14b7

Evita:

- Deadlock
- inconsistenza dei dati.

Non azzera la possibilità di starvation (un processo potrebbe accaparrarsi LOOK indefinitamente).

Operating Systems: IPC

Mutual Exclusion with Busy Waiting on Intel

- **XCHG:** `XCHG EAX, [lock].` Mette il contenuto di [lock] nel registro EAX ed inserisce un valore non nullo (es. PID) in [lock]

```
enter_region:
        MOVE REGISTER,#1          | put a 1 in the register
        XCHG REGISTER,LOCK        | swap the contents of the register and lock variable
        CMP REGISTER,#0           | was lock zero?
        JNE enter_region          | if it was non zero, lock was set, so loop
        RET                       | return to caller; critical region entered

leave_region:
        MOVE LOCK,#0              | store a 0 in lock
        RET                       | return to caller
```


D	Usando le istruzioni macchina exchange e compare_and_swap, è possibile scrivere processi che non soffrano di starvation

Operating Systems: Esercizi

Domanda 15

Quale delle seguenti affermazioni sulla memoria virtuale con paginazione è vera?

A	Il numero di bit di un indirizzo virtuale è necessariamente diverso a seconda che si usi una tabella delle pagine ad 1 o a 2 livelli
B	Il numero di bit di una entry di una tabella delle pagine di ultimo livello è uguale al numero di bit di controllo più il logaritmo (arrotondato all'intero superiore) del massimo numero di frame in memoria principale
C	Nessuna delle altre opzioni è corretta
D	Nel caso di una tabella delle pagine a 2 livelli, viene tipicamente richiesto che tutte le tabelle delle pagine di secondo livello entrino in una pagina

← Passepartout

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Esercizi

Domanda 15b1

Il numero di bit utilizzati è dato dalla architettura

Operating Systems: Memory Management

Virtual Memory: Paging 7d/7

Memory Manager: strutture avanzate per la tabella delle pagine.

Esempio di Paginazione a 2 Livelli (Forward-mapped Page Table)

Un indirizzo logico su una macchina a 32-bit con pagine di 4K è diviso in un numero di pagina di 20 bit e un offset di 12 bit

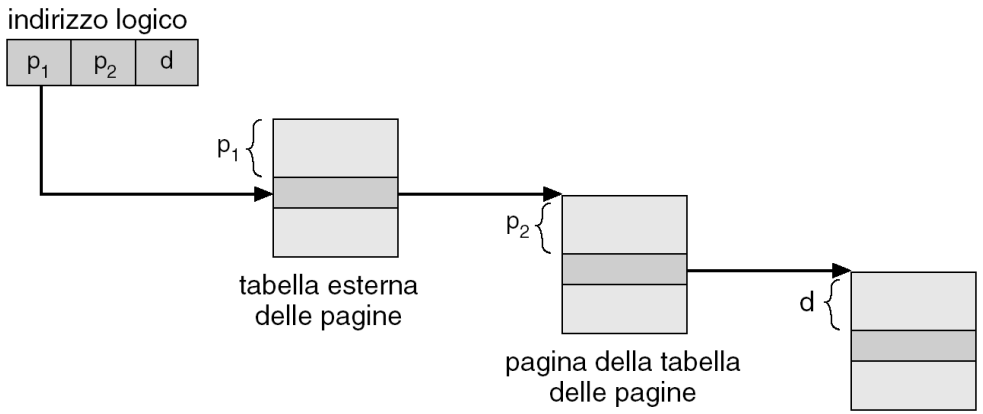
Paginazione a due livelli ➔ numero di pagina ulteriormente diviso in :

- p_1: numero di pagina da 10-bit
- p_2: spiazzamento nella pagina da 10-bit

Un indirizzo logico è diviso:

➔ p_1: indice della tabella esterna

➔ p_2: spostamento all'interno della pagina della tabella esterna



Ogni livello aggiunto implica un accesso alla memoria per la traduzione degli indirizzi ➔ raramente si superano i 3 livelli (SPARC a 32 bit)

A	Il numero di bit di un indirizzo virtuale è necessariamente diverso a seconda che si usi una tabella delle pagine ad 1 o a 2 livelli
B	Il numero di bit di una entry di una tabella delle pagine di ultimo livello è uguale al numero di bit di controllo più il logaritmo (arrotondato all'intero superiore) del massimo numero di frame in memoria principale

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti
Sapienza Università di Roma

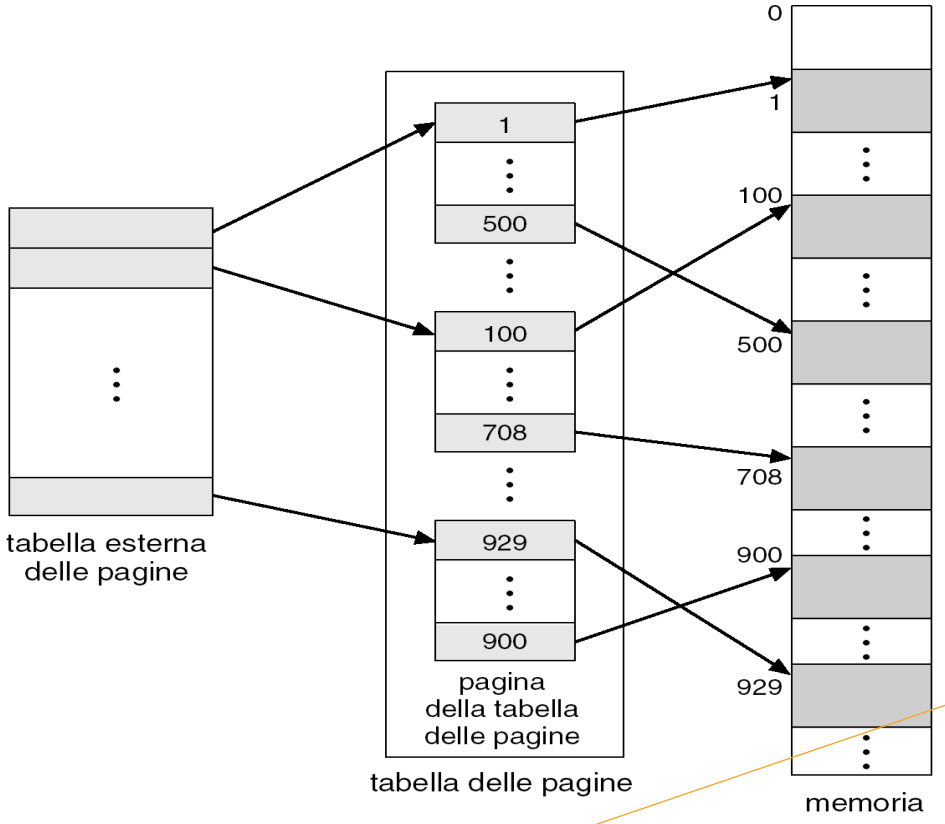
Operating Systems: Esercizi

Domanda 15b

Operating Systems: Memory Management

Virtual Memory: Paging 7c/7

Memory Manager: strutture avanzate per la tabella delle pagine.



Paginazione Gerarchica

- **Soluzione gerarchica**: suddividere lo spazio degli indirizzi logici in più tabelle di pagine (es. 2 Livelli)
- Una directory delle pagine detta tabella esterna: ogni elemento punta ad una sotto-tabella delle pagine
- Un insieme di sotto-tabelle delle pagine (Tipicamente ampia quanto una pagina al fine di essere completamente contenuta in un frame)

SAPIENZA UNIVERSITÀ DI ROMA

D	Nel caso di una tabella delle pagine a 2 livelli, viene tipicamente richiesto che tutte le tabelle delle pagine di secondo livello entrino in una pagina

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA