

Sistemi Operativi I

Corso di Laurea in Informatica

A.A. 2021-2022

Alberto Finzi

Obiettivi del Corso

Struttura e funzioni dei moderni **Sistemi Operativi**:

- Principi, componenti fondamentali, metodologie di progettazione e di sviluppo, algoritmi e strumenti di base
- Abilità di base nell'uso di una piattaforma a livello utente ed amministratore, principi di scripting e programmazione di sistema
- Particolare riferimento ai sistemi Unix e Linux

Modalità di Esame



Prova scritta

Domande aperte ed esercizi



Discussione orale

Esercizi

Discussione della prova
scritta

Programma di Massima

- **Introduzione ai Sistemi Operativi**
 - Evoluzione, strutture, architetture, componenti
- **Processi**
 - Il concetto di processo, stati dei processi, funzioni del kernel, algoritmi di schedulazione, sincronizzazione dei processi e deadlock
- **Memoria**
 - Gestione memoria principale, swapping, partizione, segmentazione e paginazione, memoria virtuale
- **Sistemi I/O**
 - Architetture e dispositivi di I/O, interfacce, sottosistema per l'I/O del nucleo, etc.
- **Dati permanenti**
 - File, metodi di allocazione, directory e metodi di accesso, file system, etc.
- **Cenni su Sistemi distribuiti**

Argomenti Rilevanti

- Introduzione ai Sistemi Operativi
 - Evoluzione, strutture, architetture, componenti
 - Concetto di multiprogrammazione e multiutente
 - Concetto di Nucleo
 - modalità operativa duale, bit di modalità, etc.
 - Gestione interruzioni
 - Gerarchie di memorie
 - Servizi dei Sistemi Operativi
 - Chiamate di Sistema
 - esercizi, vedi dopo
 - Interfaccia utente
 - Bash shell e comandi (esercizi vedi dopo)
 - Struttura dei Sistemi Operativi
 - Monolitici, stratificati, microkernel, modulari, ibridi

Esempi domande

- Esempio domande:
 - **Spiegare la funzione della modalità operativa duale (kernel mode e user mode)**
 - **Cos'è il bit di modalità e a cosa serve?**

Esempi domande

- Sistemi multiprogrammati e multiutente richiedono **meccanismi di protezione**
- Le operazioni in dual-mode permettono al SO di proteggersi e proteggere altri componenti
- Si distinguono due modalità: **user mode** e **kernel mode** (modo utente e supervisore)
- Il **bit di modalità** (hardware) permette di distinguere quando il sistema esegue codice utente o kernel
 - Alcune istruzioni sono privilegiate quindi solo eseguibili in modalità kernel
 - Con interrupt/trap si va alla modalità kernel, quando viene servita l'interruzione il sistema ripristina la modalità user

Esempi domande

- Esempio domande:
 - **Spiegare la differenza tra interrupt e polling**
 - **Descrivere il ciclo di I/O orientato alle interruzioni**

Esempi domande

- Esempio domande:
 - **Quali sono i vantaggi e gli svantaggi di un approccio monolitico e stratificato al progetto di un SO?**
 - **Qual è il principale vantaggio dell'approccio microkernel nella progettazione di un SO?**

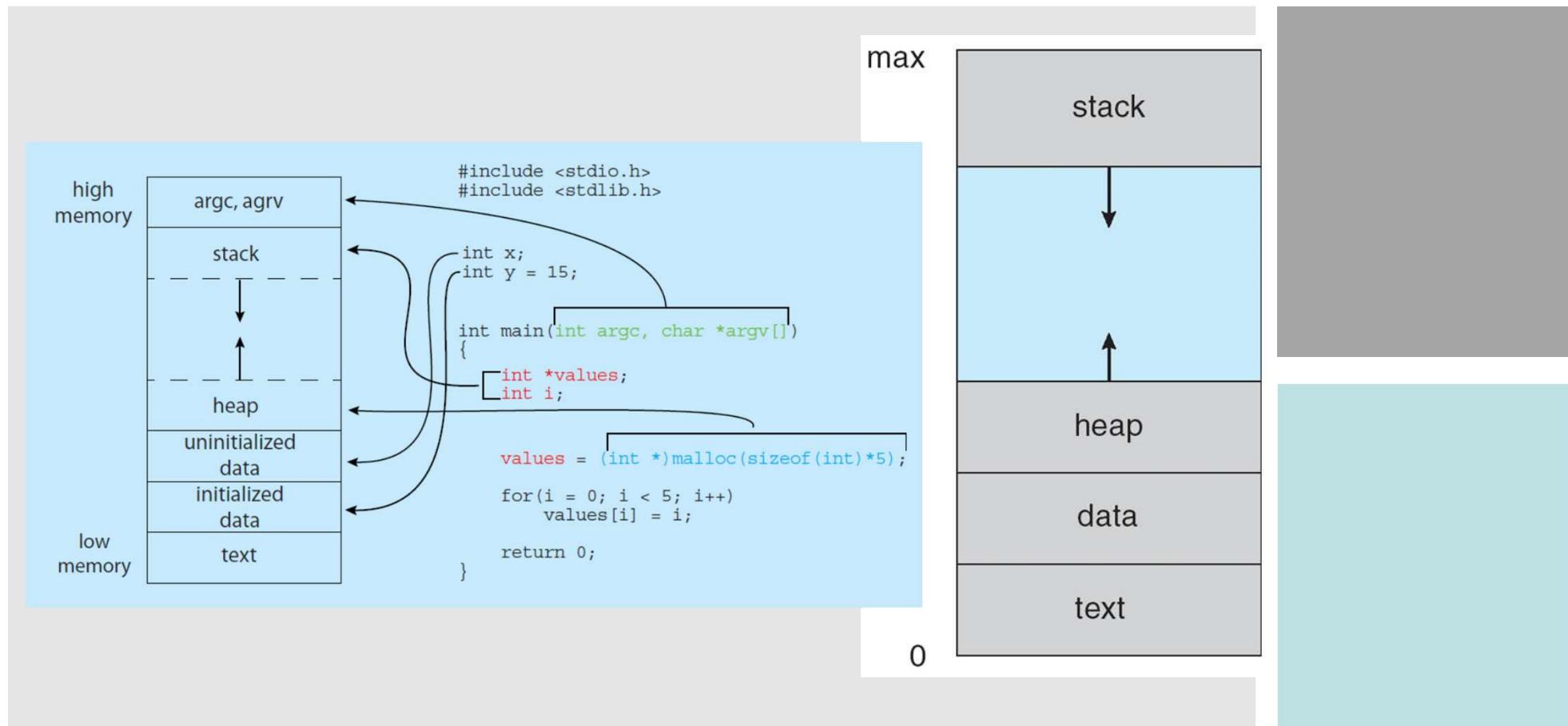
Argomenti Rilevanti

- Processi
 - Definizione di processo
 - Allocazione processo in memoria
 - Stati di un processo
 - Descrittore di processo
 - Commutazione tra processi
 - Schedulazione di processi
 - Diagrammi di accodamento
 - Schedulazione di breve/lugo/medio
 - Operazioni su processi e chiamate di Sistema
 - fork, wait, exec, exit (esercizi)
 - Comunicazione interprocesso
 - Message passing e shared memory
 - Problema produttori consumatori
 - IPC POSIX (esercizi)
 - pipe, mmap, socket

Esempi domande

- Esempio domande:
 - **Descrivere brevemente l'allocazione di memoria di un processo**

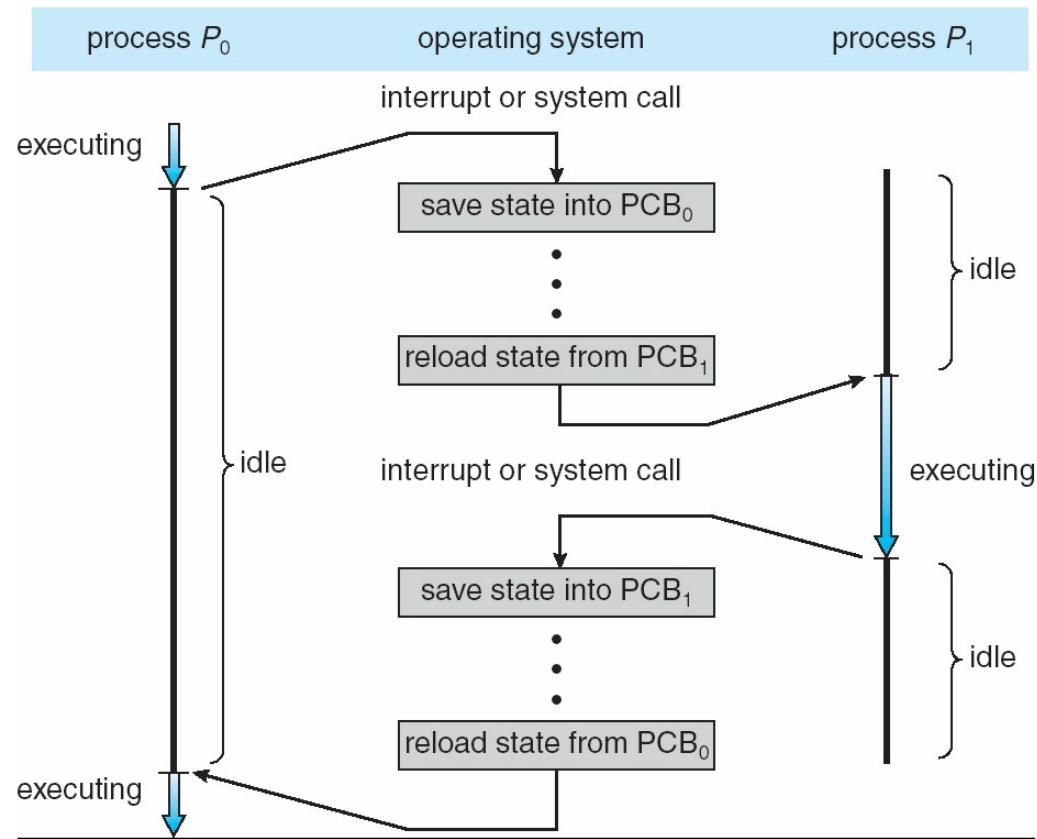
Esempi domande



Esempi domande

- Esempio domande:
 - **Descrivere brevemente il context-switch tra processi**

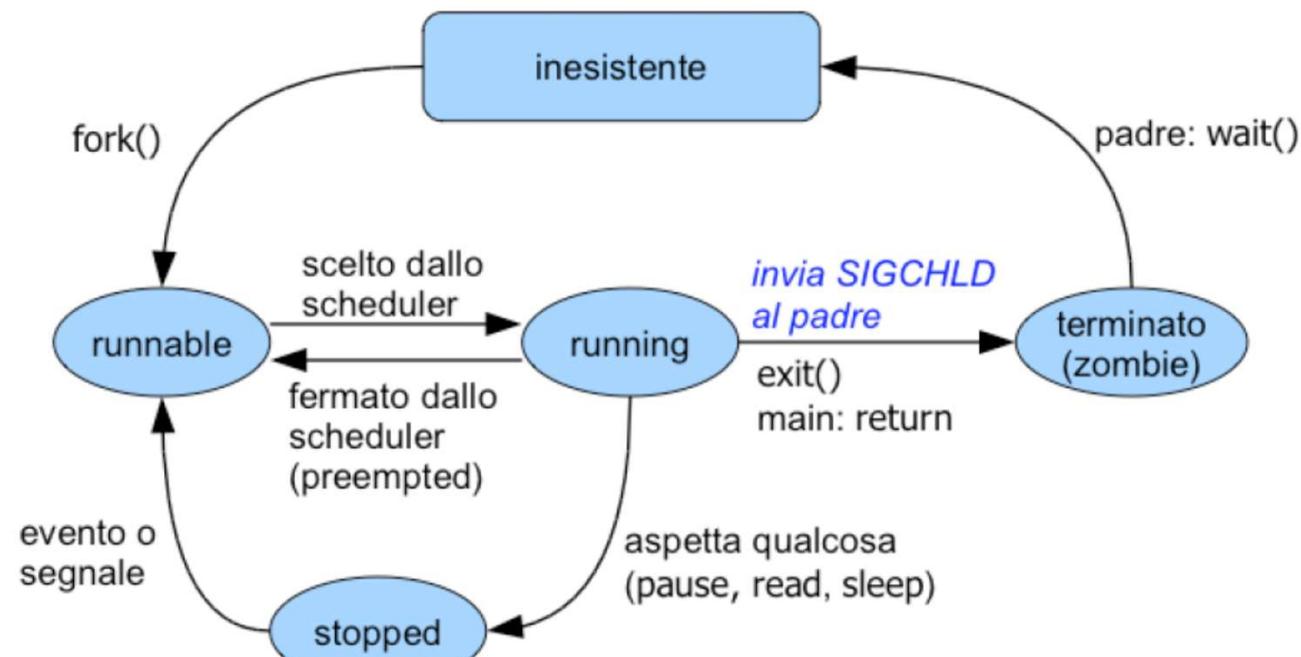
Esempi domande



Esempi domande

- Esempio domande:
 - **Quale delle seguenti affermazioni sui processi zombie è vera**
 - Ogni terminazione di un processo fa diventare processi zombie tutti i figli di quel processo.
 - Un processo zombie continua a consumare le risorse che aveva impiegato
 - Un processo rimane zombie finchè non è chiamata una wait

Esempi domande



Esempi domande

- Esempio domande:
 - **L'invocazione di una chiamata di sistema fork() implica che (sono possibili più risposte):**
 - Padre e Figlio hanno un identico, ma separato spazio di indirizzamento
 - Il Figlio prende lo stesso PID del padre
 - Tutte le variabili inizializzate prima della fork hanno lo stesso valore per entrambi i processi
 - La modifica di una variabile ha effetto su entrambi gli spazi di indirizzamento

Esempi domande

- Esempio domande:

- **Quando un processo crea un nuovo processo con una fork() quale dei seguenti segmenti sono condivisi tra il processo padre e figlio?**
 - a. Stack
 - b. Heap
 - c. Segmenti di shared memory

Esempi domande

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main()
{
pid_t pid;

    pid = fork();

    if (pid == 0) { /* child process */
        value += 15;
        return 0;
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d",value); /* LINE A */
        return 0;
    }
}
```

- Esempio domande:
- Output linea A?

Esempi domande

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```

- Esempio domande:
- Quanti processi sono creati?

Esempi domande

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```

- Esempio domande:
- Quanti processi sono creati?
- 2^3

Esempi domande

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    int i;

    for (i = 0; i < 4; i++)
        fork();

    return 0;
}
```

- Esempio domande:
- Quanti processi sono creati?

Esempi domande

```
int main()
{
pid_t pid;

/* fork a child process */
pid = fork();

if (pid < 0) { /* error occurred */
    fprintf(stderr, "Fork Failed");
    return 1;
}
else if (pid == 0) { /* child process */
    execlp("/bin/ls","ls",NULL);
    printf("LINE J");
}
else { /* parent process */
    /* parent will wait for the child to complete */
    wait(NULL);
    printf("Child Complete");
}

return 0;
}
```

- Esempio domande:
- Quando si raggiunge la linea J

Argomenti Rilevanti

- Thread e concorrenza
 - Definizione di thread
 - thread vs processo
 - Multicore programming
 - Concorrenza vs Parallelismo
 - Legge di Amdahl (esercizi)
 - User thread e kernel thread
 - Modelli multithreading: Many-to-One, One-to-One, Many-to-Many, Two level model
 - Librerie thread
 - Thread Linux/Windows/Java
 - Pthreads (esercizi)
 - Creazione, cancellazione
 - thread specific data
 - Threading隐式
 - Problematiche thread
 - exec e thread, thread e segnali, cancellazione thread

Esempi domande

- Esempio domande:
- **Quali delle seguenti componenti dello stato del programma sono condivisi tra thread in un processo multithreaded?**
 - a. Register values
 - b. Heap memory
 - c. Global variables
 - d. Stack memory

Esempi domande

- Esempio domande:
- **È possibile aver concorrenza senza parallelismo?**
Spiegare brevemente

Esempi domande

- Esempio domande:
- **Usando la legge di Amdahl calcolare il guadagno in velocità di un'applicazione che ha il 67 percento di componente parallela con (a) due core e (b) quattro core**

Esempi domande

- Usando la legge di Amdahl calcolare il guadagno in prestazioni di un'applicazione che ha il 40 percento di componente parallela con (a) otto core e (b) sedici core

$$speedup \leq \frac{1}{S + \frac{(1-S)}{N}}$$

P = 40% = 0,4 S = 60% = 0,6

N = 8 speedup = 1,53

N = 16 speedup = 1,60

Esempi domande

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
pid_t pid;
pthread_t tid;
pthread_attr_t attr;

pid = fork();

if (pid == 0) { /* child process */
pthread_attr_init(&attr);
pthread_create(&tid,&attr,runner,NULL);
pthread_join(tid,NULL);
printf("CHILD: value = %d",value); /* LINE C */
}
else if (pid > 0) { /* parent process */
wait(NULL);
printf("PARENT: value = %d",value); /* LINE P */
}
}

void *runner(void *param) {
value = 5;
pthread_exit(0);
}
```

Cosa stampa in
linea C e P?

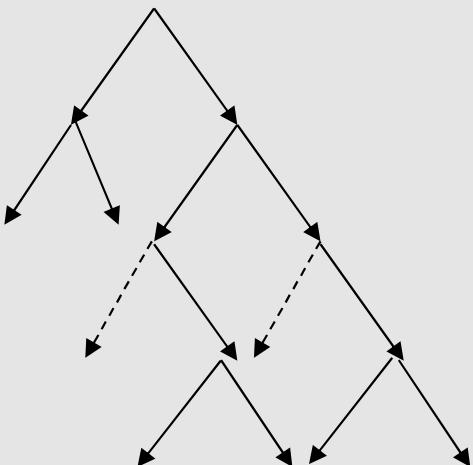
Esempi domande

```
pid_t pid;  
  
pid = fork();  
if (pid == 0) { /* child process */  
    fork();  
    thread_create( . . . );  
}  
fork();
```

- Quanti processi e quanti thread?

Esempi domande

```
pid_t pid;  
  
pid = fork();  
if (pid == 0) { /* child process */  
    fork();  
    thread_create( . . . );  
}  
fork();
```



- 6 processi e 2 thread

Argomenti Rilevanti

- Scheduling CPU
 - **Concetti di base**
 - CPU brust, I/O brust
 - Criteri di scheduling
 - Prelazione
 - dispatcher
 - **Tipi di scheduling**
 - FCFS, SJF, shortest-remaining-time-first, scheduling con priorità, Round Robin, Round Robin con priorità, Code Multiple, Code Multiple con Feedback
 - Esercizi
 - **Thread scheduling**
 - **Scheduling multiprocessore**
 - Concetti e problematiche
 - **Real-time scheduling**
 - Task periodici (rate monotonic scheduling)
 - Earliest deadline first
 - Esercizi
 - Esempi sistemi operativi
 - Linux scheduling, windows scheduling
 - Valutazione algoritmi di scheduling

Esempi domande

Si considerino i processi in tabella con tempi di esecuzione (in millisecondi) e le priorità indicate (più piccola più alta)

<i>Processo no.</i>	<i>Tempo di esecuzione</i>	<i>Priorità</i>
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Arrivano tutti al tempo 0 con ordine di arrivo (P_1, P_2, P_3, P_4, P_5),

- Si illustri l'ordine di esecuzione ed i tempi di attesa medi per SJF e a priorità

Esempi domande

- **Utilizzo CPU** –percentuale di CPU utilizzata; occorre mantenere la CPU occupata.
- **Throughput** – numero di processi che completano l'esecuzione per unità di tempo
- **Turnaround time** – tempo di completamento di processo
 - tempo totale per eseguire un processo
 - accesso memoria + coda ready + CPU + I/O
- **Waiting time** – tempo di attesa per un processo nella coda ready
- **Response time** – tempo che intercorre tra una richiesta e la prima risposta
 - Importante per sistemi interattivi, quando l'elaborazione continua dopo un output
 - Alternativa al tempo di turnaround che prevede l'esecuzione completa

Esempi domande

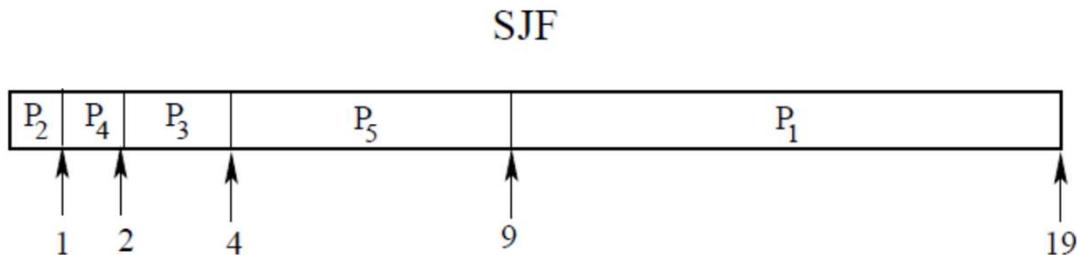
- Esercizi scheduling

<i>Processo no.</i>	<i>Tempo di esecuzione</i>	<i>Priorità</i>
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Soluzione

SJF

$$t_a(SJF) = \frac{9 + 0 + 2 + 1 + 4}{5} = 3.2$$



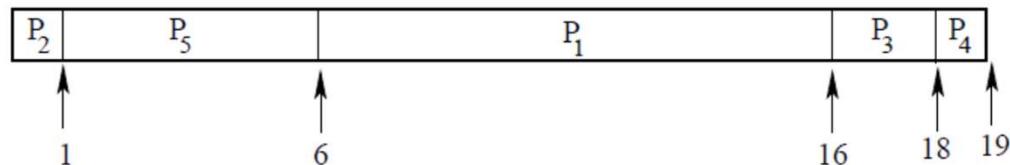
Esempi domande

- Esercizi scheduling

<i>Processo no.</i>	<i>Tempo di esecuzione</i>	<i>Priorità</i>
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Soluzione

Priorità



$$t_a(Pr) = \frac{6 + 0 + 16 + 18 + 1}{5} = 8.2$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
<i>A</i>	0	3
<i>B</i>	2	6
<i>C</i>	4	4
<i>D</i>	6	5
<i>E</i>	8	2

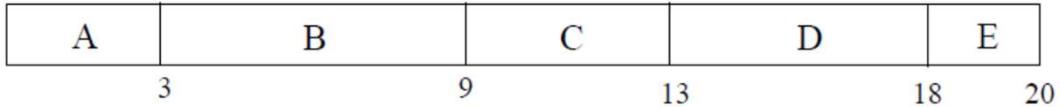
Si calcoli il tempo medio di attesa e di turnaround per FCFS e RR con quanti di tempo 1 e 4 e SJF

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

processo	tempo di arrivo	CPU-burst (millisec.)
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Si calcoli il tempo medio di attesa per FCFS

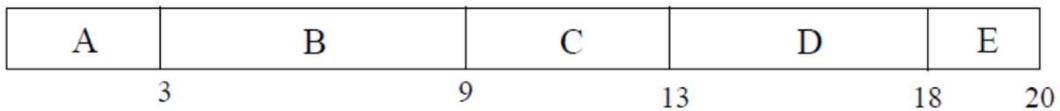
$$t_a(FCFS) = \frac{(3 - 3 - 0) + (9 - 6 - 2) + (13 - 4 - 4) + (18 - 5 - 6) + (20 - 2 - 8)}{5} = 4.6$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

processo	tempo di arrivo	CPU-burst (millisec.)
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Si calcoli il tempo medio di turnaround per FCFS

$$t_{tr}(FCFS) = \frac{(3-0)+(9-2)+(13-4)+(18-6)+(20-8)}{5} = 8.6$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
<i>A</i>	0	3
<i>B</i>	2	6
<i>C</i>	4	4
<i>D</i>	6	5
<i>E</i>	8	2

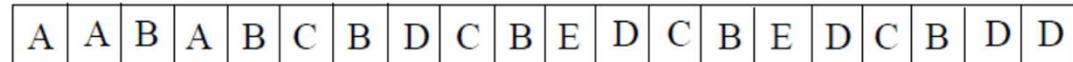
Si calcoli il tempo medio di attesa
RR quanto = 1

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Si calcoli il tempo medio di attesa
RR quanto = 1

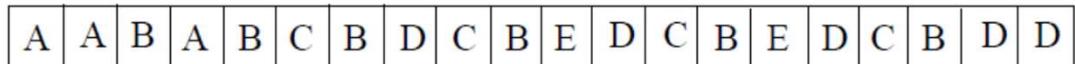
$$t_a(RR - 1) = \frac{(4 - 3 - 0) + (18 - 6 - 2) + (17 - 4 - 4) + (20 - 5 - 6) + (15 - 2 - 8)}{5} = 6.8$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



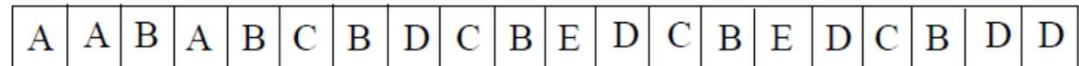
Si calcoli il tempo medio di turnaround
RR quanto = 1

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Si calcoli il tempo medio di turnaround RR quanto = 1

$$t_{tr}(RR - 1) = \frac{(4 - 0) + (18 - 2) + (17 - 4) + (20 - 6) + (15 - 8)}{5} = 10.8$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
<i>A</i>	0	3
<i>B</i>	2	6
<i>C</i>	4	4
<i>D</i>	6	5
<i>E</i>	8	2

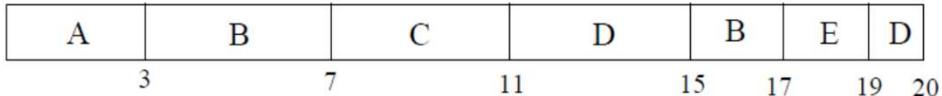
Si calcoli il tempo medio di attesa e turnaround RR quanto = 4

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



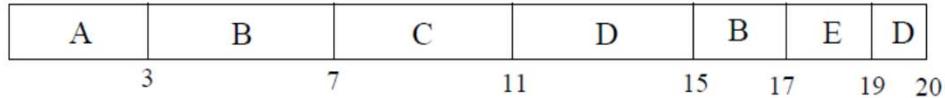
Si calcoli il tempo medio di attesa e turnaround RR quanto = 4

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Si calcoli il tempo medio di attesa e turnaround RR quanto = 4

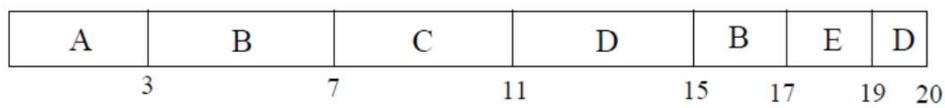
$$t_a(RR - 4) = \frac{(3 - 3 - 0) + (17 - 6 - 2) + (11 - 4 - 4) + (20 - 5 - 6) + (19 - 2 - 8)}{5} = 6$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Si calcoli il tempo medio di attesa e turnaround RR quanto = 4

$$t_{tr}(RR - 4) = \frac{(3 - 0) + (17 - 2) + (11 - 4) + (20 - 6) + (19 - 8)}{5} = 10$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
<i>A</i>	0	3
<i>B</i>	2	6
<i>C</i>	4	4
<i>D</i>	6	5
<i>E</i>	8	2

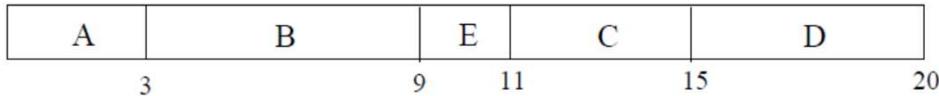
Si calcoli il tempo medio di attesa e turnaround SJF

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
<i>A</i>	0	3
<i>B</i>	2	6
<i>C</i>	4	4
<i>D</i>	6	5
<i>E</i>	8	2



Si calcoli il tempo medio di attesa e turnaround SJF

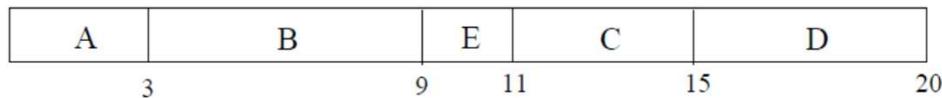
$$t_a(SJF) = \frac{(3 - 3 - 0) + (9 - 6 - 2) + (15 - 4 - 4) + (20 - 5 - 6) + (11 - 2 - 8)}{5} = 3.6$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e CPU-brust indicato

<i>processo</i>	<i>tempo di arrivo</i>	<i>CPU-burst (millisec.)</i>
<i>A</i>	0	3
<i>B</i>	2	6
<i>C</i>	4	4
<i>D</i>	6	5
<i>E</i>	8	2



Si calcoli il tempo medio di attesa e turnaround SJF

$$t_{tr}(SJF) = \frac{(3 - 0) + (9 - 2) + (15 - 4) + (20 - 6) + (11 - 8)}{5} = 7.6$$

Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e tempo di esecuzione

<i>processo</i>	<i>tempo di arrivo</i>	<i>tempo di esecuzione</i>
P_1	0	20
P_2	8	5
P_3	3	12
P_4	10	6
P_5	7	8

Calcolare tempo medio di attesa e turnaround per SJF preemptive

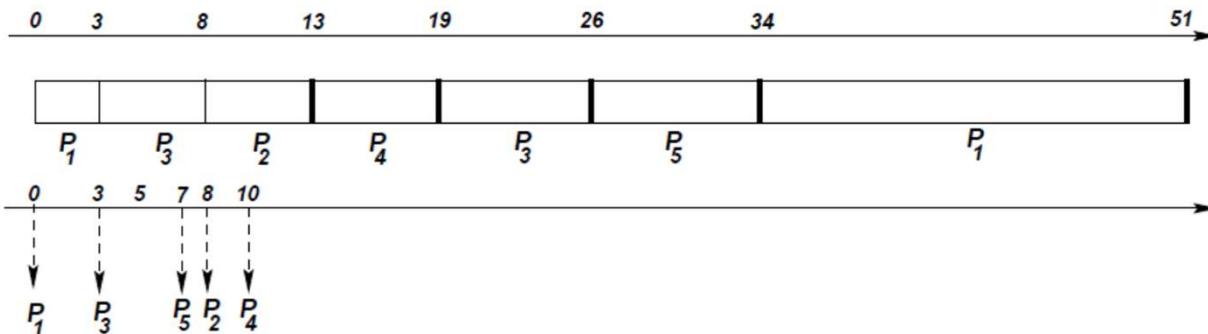
Esempi domande

- Esercizi scheduling

Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e tempo di esecuzione

processo	tempo di arrivo	tempo di esecuzione
P_1	0	20
P_2	8	5
P_3	3	12
P_4	10	6
P_5	7	8

Calcolare tempo medio di attesa e turnaround per SJF preemptive



Esempi domande

- Esercizi scheduling

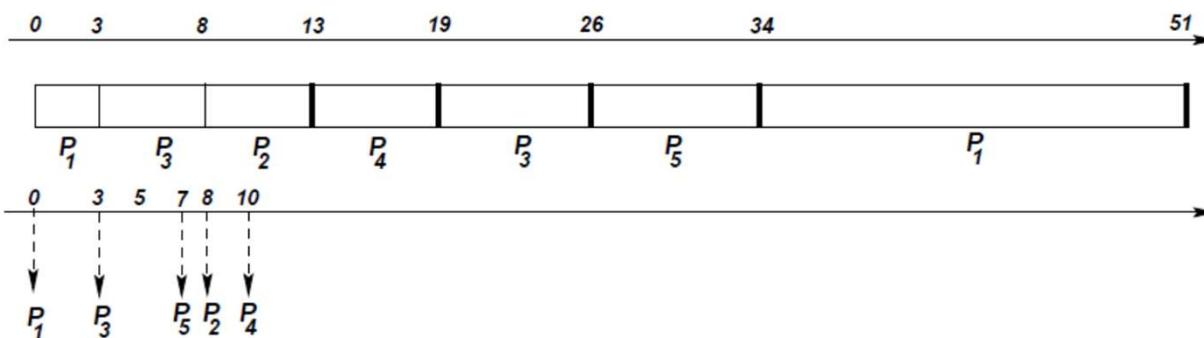
Si considerino i processi in tabella con tempi di arrivo (in millisecondi) e tempo di esecuzione

processo	tempo di arrivo	tempo di esecuzione
P_1	0	20
P_2	8	5
P_3	3	12
P_4	10	6
P_5	7	8

Calcolare tempo medio di attesa e turnaround per SJF preemptive

$$t_a = \frac{31 + 0 + 11 + 3 + 19}{5} = 12.8 \text{ msec}$$

$$t_{tr} = \frac{51 + 5 + 23 + 9 + 27}{5} = 23 \text{ msec}$$



Esempi domande

- Esercizi scheduling

Si considerino due processi P1 e P2 con
 $p_1 = 50$, $t_1 = 25$,
 $p_2 = 75$, $t_2 = 30$

0 p1 25; 25 p2 50; 50 p1 75; 75 p2 80

Possano essere schedulati con
rate-monotonic scheduling?

Earliest deadline?

Esempi domande

- Esercizi scheduling

Quali dei seguenti algoritmi possono portare a starvation:

1. FCFS
2. SJF
3. RR
4. Priority based

Argomenti Rilevanti

- Sincronizzazione
 - Concetto di corsa critica
 - Problema della sezione critica
 - requisiti
 - Gestione della corsa critica
 - Soluzione di Peterson
 - Hardware per sincronizzazione
 - Lock, barriere, istruzioni atomiche
 - Test and set, compare and swap
 - Mutex, semafori
 - Esercizi
 - Deadlock starvation
 - Esercizi
 - Monitor e variabili di condizione
 - Esercizi
 - Implementazione monitor
 - Inversione di priorità
 - Concetto, esempio

Esercizi

```
wait(mutex);  
...  
critical section  
...  
wait(mutex);
```

- Descrivi cosa accade in questo frammento di codice

Esercizi

```
push(item) {  
    if (top < SIZE) {  
        stack[top] = item;  
        top++;  
    }  
    else  
        ERROR  
}  
  
pop() {  
    if (!is_empty()) {  
        top--;  
        return stack[top];  
    }  
    else  
        ERROR  
}  
  
is_empty() {  
    if (top == 0)  
        return true;  
    else  
        return false;  
}
```

- Date le implementazione di push, pop, is_empty per una pila basata su array
- descrivi su quali dati si può generare una race condition e come evitarlo usando mutex lock

Argomenti Rilevanti

- Esempi di Sincronizzazione
 - **Bounded buffer**
 - **Lettori scrittori**
 - **Cena dei filosofi**
- Sincronizzazione in Solaris, Linux, Windows
 - **POSIX mutex e variabili di condizione**
 - Esercizi

Esempio Esercizi

- Scrivere lo pseudocodice della soluzione del problema lettori/scrittori, prima variante

Esempio Esercizi

□ Struttura scrittore

```
do {  
    wait(rw_mutex);  
    ...  
    /* writing is performed */  
    ...  
    signal(rw_mutex);  
} while (true);
```

Esempio Esercizi

□ Struttura del lettore

```
do {
    wait(mutex);
    read_count++;
    if (read_count == 1)
        wait(rw_mutex);
    signal(mutex);

    ...
    /* reading is performed */

    ...

    wait(mutex);
    read_count--;
    if (read_count == 0)
        signal(rw_mutex);
    signal(mutex);
} while (true);
```

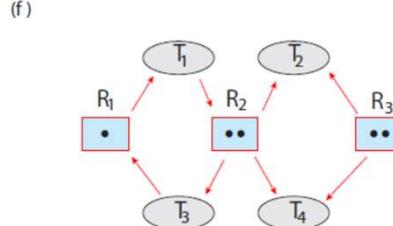
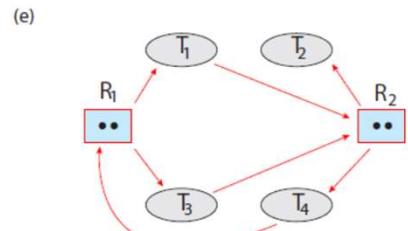
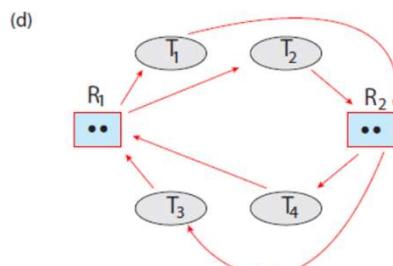
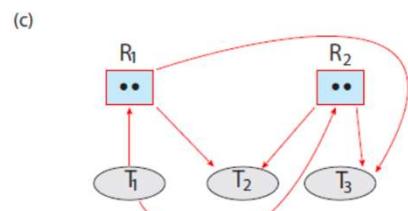
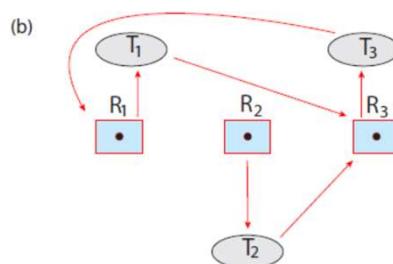
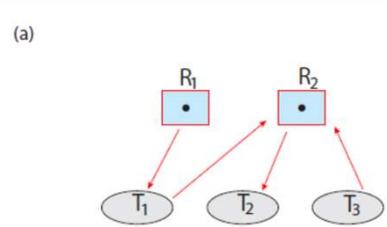
Argomenti Rilevanti

- Deadlock
 - Concetti principali
 - Caratterizzazione deadlock
 - Grafo di allocazione di risorse
 - Metodi di gestione del deadlock
 - Prevention, avoidance
 - Algoritmo del grafo delle risorse
 - Algoritmo del banchiere
 - Detection
 - Recovery

Esempio Esercizi

- Scrivere lo pseudocodice dell'algoritmo del banchiere utilizzato nell'ambito della prevenzione dei deadlock

Esempio Esercizi



- In quale situazione c'è deadlock?

Esempio Esercizi

- Esempio: 12 risorse e 3 thread

	<u>Maximum Needs</u>	<u>Current Needs</u>
T_0	10	5
T_1	4	2
T_2	9	2

- Il Sistema è in stato sicuro
 - Allocate 9 ne rimangono 3
 - La sequenza $\langle T_1, T_0, T_2 \rangle$ soddisfa il requisite
 - T1 ne prende 2 e restituisce 4
 - T0 ne prende 5 e restituisce 10
 - T2 ne prende 7 e finisce
 - Se invece si alloca a T2 un'ulteriore risorsa al tempo t1 lo stato non è più safe

Esempio Esercizi

	<i>Allocation</i>				<i>Max</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
T_0	2	1	0	6	6	3	2	7
T_1	3	3	1	3	5	4	1	5
T_2	2	3	1	2	6	6	1	4
T_3	1	2	3	4	4	3	4	5
T_4	3	0	3	0	7	2	6	1

- Considera la seguente situazione e utilizza l'algoritmo del banchiere per rispondere
 - Cosa contiene la matrice NEED?

Esempio Esercizi

	<i>Allocation</i>				<i>Max</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
T_0	2	1	0	6	6	3	2	7
T_1	3	3	1	3	5	4	1	5
T_2	2	3	1	2	6	6	1	4
T_3	1	2	3	4	4	3	4	5
T_4	3	0	3	0	7	2	6	1

$$\text{Need } [i,j] = \text{Max}[i,j] - \text{Allocation } [i,j]$$

- Considera la seguente situazione e utilizza l'algoritmo del banchiere per rispondere
 - Cosa contiene la matrice NEED?

Esempio Esercizi

	<i>Allocation</i>				<i>Max</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
T_0	2	1	0	6	6	3	2	7
T_1	3	3	1	3	5	4	1	5
T_2	2	3	1	2	6	6	1	4
T_3	1	2	3	4	4	3	4	5
T_4	3	0	3	0	7	2	6	1

- Considera la seguente situazione e utilizza l'algoritmo del banchiere per rispondere
 - Cosa contiene la matrice NEED?
 - (4,2,2,1),
(2,1,0,2),
(4,3,0,2),
(3,1,1,1),
(4,2,3,1)

Esempio Esercizi

	<i>Allocation</i>				<i>Max</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
T_0	1	2	0	2	4	3	1	6
T_1	0	1	1	2	2	4	2	4
T_2	1	2	4	0	3	6	5	1
T_3	1	2	0	1	2	6	2	3
T_4	1	0	0	1	3	1	1	2

- a. $\text{Available} = (2, 2, 2, 3)$
- b. $\text{Available} = (4, 4, 1, 1)$

- Considera la seguente situazione e utilizza l'algoritmo del banchiere per rispondere
- Determina se lo stato è safe o unsafe

Esempio Esercizi

	<i>Allocation</i>				<i>Max</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
T_0	1	2	0	2	4	3	1	6
T_1	0	1	1	2	2	4	2	4
T_2	1	2	4	0	3	6	5	1
T_3	1	2	0	1	2	6	2	3
T_4	1	0	0	1	3	1	1	2

a. $\text{Available} = (2, 2, 2, 3)$

b. $\text{Available} = (4, 4, 1, 1)$

a. Si, un ordinamento possibile è $T_4 T_0 T_1 T_2 T_3$

b. Si, un ordinamento possibile è $T_2 T_4 T_3 T_1 T_0$

- Considera la seguente situazione e utilizza l'algoritmo del banchiere per rispondere
- Determina se lo stato è safe o unsafe

Esempio Esercizi

	<u>Allocation</u>	<u>Max</u>	<u>Available</u>
	A B C D	A B C D	A B C D
T_0	0 0 1 2	0 0 1 2	1 5 2 0
T_1	1 0 0 0	1 7 5 0	
T_2	1 3 5 4	2 3 5 6	
T_3	0 6 3 2	0 6 5 2	
T_4	0 0 1 4	0 6 5 6	

- Contenuto matrice Need
- Determina se lo stato è safe o unsafe
- La richiesta (0, 4, 2, 0) può essere immediatamente garantita?

Argomenti Rilevanti

- Memoria Principale
 - Concetti principali
 - Binding degli indirizzi
 - Indirizzi fisici e logici
 - Memory Management Unit
 - Caricamento dinamico
 - Protezione
 - Allocazione
 - Algoritmi ed Esercizi
 - Frammentazione interna/esterna
 - concetti, esempi, esercizi
 - Metodi per il Paging
 - Metodi ed Esercizi
 - **Swapping (concetti)**
 - **Segmentazione (concetti)**

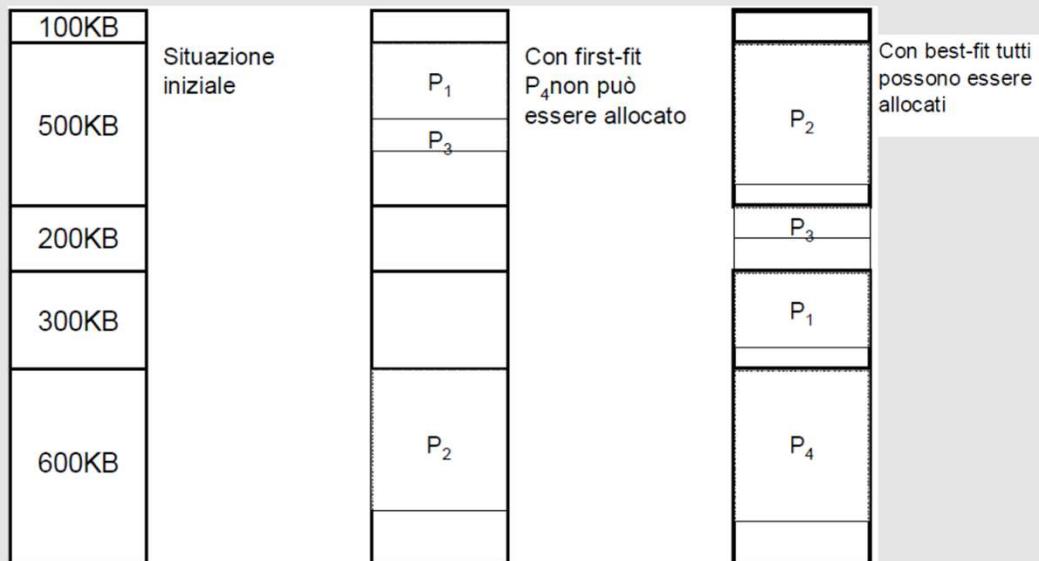
Esercizi

Date le seguenti partizioni di memoria: 100KB, 500KB, 200KB, 300KB e 600KB, descrivere come sono disposti dagli algoritmi di *first-fit* e *best-fit* i processi di 212KB, 417KB, 112KB e 426KB (in ordine).

- Esercizio Memoria Principale

Esercizi

Date le seguenti partizioni di memoria: 100KB, 500KB, 200KB, 300KB e 600KB, descrivere come sono disposti dagli algoritmi di *first-fit* e *best-fit* i processi di 212KB, 417KB, 112KB e 426KB (in ordine).



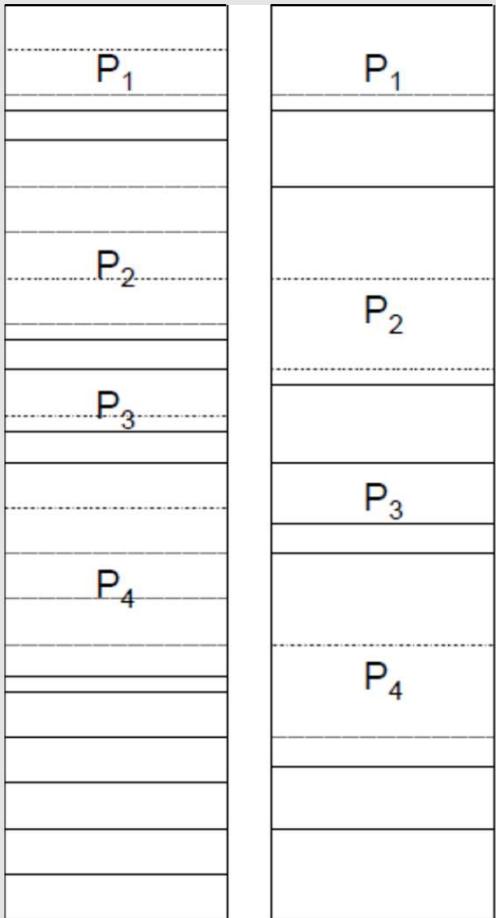
- Esercizio Memoria Principale

Argomenti Rilevanti

Se la memoria è 2000KB e viene gestita mediante la paginazione, con pagine da 100KB e 200KB, si descriva l'allocazione della memoria, relativamente a processi di dimensioni 212KB, 417KB, 112KB e 426KB.

- Esercizio Memoria Principale

Esercizio



- La frammentazione interna è maggiore nel caso di pagine più grandi.
- La memoria libera è di 5 blocchi (500K) nel primo caso, uno solo (200K) nel secondo.

Esercizi

Considerate la seguente tabella dei segmenti:

Segmento	Base	Lunghezza
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Calcolate gli indirizzi fisici corrispondenti ai seguenti indirizzi logici:

- a) < 0, 430 >
- b) < 1, 10 >
- c) < 2, 500 >
- d) < 3, 400 >
- e) < 4, 112 >

- Esercizio Memoria Principale

Esercizi

Considerate la seguente tabella dei segmenti:

Segmento	Base	Lunghezza
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

- Calcolate gli indirizzi fisici corrispondenti ai seguenti indirizzi logici:

- a) < 0, 430 > → $219 + 430 = 649$
- b) < 1, 10 > → $2300 + 10 = 2310$
- c) < 2, 500 > → **500 > 100 errore di segmentazione**
- d) < 3, 400 > → $1327 + 400 = 1727$
- e) < 4, 112 > → **112 > 96 errore di segmentazione**

- Esercizio Memoria Principale

Esercizi

Un programma ha il segmento di testo di 200KB, quello dei dati di 100KB e uno stack di 50KB. Se la memoria è organizzata a pagine da 16KB:

1. Quanto è lunga la tabella delle pagine del processo?
2. Quanta memoria può al più essere gestita se si hanno a disposizione 32 bit per l'indirizzamento?

- Esercizio Memoria Principale

Esercizi

Un programma ha il segmento di testo di 200KB, quello dei dati di 100KB e uno stack di 50KB. Se la memoria è organizzata a pagine da 16KB:

1. Quanto è lunga la tabella delle pagine del processo?
2. Quanta memoria può al più essere gestita se si hanno a disposizione 32 bit per l'indirizzamento?

1. Risulta:

- Testo 200KB → $200/16 = 13$ pagine
- Dati 100KB → $100/16 = 7$ pagine
- Stack 50KB → $50/16 = 4$ pagine

Per un totale di **24 pagine**

2. Per indirizzare pagine da 16KB, c'è bisogno ($16K=2^{14}$) di 14 bit. Restano 18 bit per generare i numeri delle pagine → 256K pagine, pari a 4GB di memoria complessiva.

- Esercizio Memoria Principale

Esercizi

Date pagine da 1 KB quali sono i numeri di pagina e offset dei seguenti indirizzi?

- a. 21205
- b. 164250
- c. 121357
- d. 16479315
- e. 27253187

- Esercizio Memoria Virtuale

Esercizi

Date pagine da 1 KB quali sono i numeri di pagina e offset dei seguenti indirizzi?

- a. 21205
- b. 164250
- c. 121357
- d. 16479315
- e. 27253187

- a. 20 offset 725
- b. 160 offset 410
- c. 118 offset 525
- d. 16093 offset 83
- e. 26614 offset 451

- Esercizio Memoria

Argomenti Rilevanti

- Memoria Virtuale
 - Concetti principali
 - Spazio degli indirizzi virtuali
 - Demand paging
 - Page fault e instruction restart
 - Frame liberi
 - Performance
 - Ottimizzazioni
 - copy-on-write
 - Sostituzione delle pagine (esercizi)
 - Thrashing
 - Locality model (esercizi)
 - Modello working set
 - **Memoria Kernel**
 - **Buddy e Slab**
 - **Linux, Windows, Solaris**

Esercizi

- Un computer ha indirizzi a 32 bit, e pagine su due livelli. Gli indirizzi virtuali sono composti da:
 - ☞ 9 bit per il numero della pagina principale
 - ☞ 11 bit per il numero della pagina di secondo livello
 - ☞ 12 bit per l'offset nella pagina
- Quanto sono grandi le pagine e quante pagine virtuali ci sono?

- Esercizio Memoria Virtuale

Esercizi

- Un computer ha indirizzi a 32 bit, e pagine su due livelli. Gli indirizzi virtuali sono composti da:
 - ☞ 9 bit per il numero della pagina principale
 - ☞ 11 bit per il numero della pagina di secondo livello
 - ☞ 12 bit per l'offset nella pagina
- Quanto sono grandi le pagine e quante pagine virtuali ci sono?

- Numero di pagine: $2^9 \times 2^{11} = 2^{20} \approx 10^6$
- Dimensione di ciascuna pagina: $2^{12} = 4 \times 2^{10} = 4K$

- Esercizio Memoria Virtuale

Esercizi

- Si consideri un processo A con 8 pagine logiche di 1K, e le seguenti configurazioni della tabella delle pagine

	Presente – Assente	P Fisica
0	1	435
1	0	0
2	1	171
3	0	0
4	1	248
5	1	732
6	0	0
7	1	222

e del TLB (memoria associativa) di 4 posizioni

	Valido	Pagina Virtuale	Pagina Fisica
0	1	2	171
1	1	4	248
2	0	0	0
3	0	0	0

Discutere cosa accade quando A cerca di accedere alle seguenti parole:

- a) 400
- b) 2100
- c) 1050

- Esercizio Memoria Virtuale

Esercizi

- Si accede alla pagina 0, che non è presente nel TLB ma è in memoria, quindi il descrittore della pagina 0 viene caricato nel TLB nella posizione 2.
- Si accede alla pagina logica 2 il cui descrittore si trova nel TLB, quindi la traduzione dell'indirizzo non richiede modifiche del TLB
- Si accede alla pagina logica 1 che non è presente nel TLB e non è presente in memoria, verrà pertanto generato un page fault, dopodiché il corrispondente descrittore di pagina verrà caricato nel TLB.

- Esercizio Memoria Virtuale

Esercizi

Spazio indirizzi logici 2048 pagine con pagine da 4 KB mappate su memoria fisica di 512 frame

Quanti bit per indirizzo logico?
Quanti bit per indirizzo fisico?

- Esercizio Memoria

Esercizi

Spazio indirizzi logici 2048 pagine con pagine da 4 KB mappate su memoria fisica di 512 frame

Quanti bit per indirizzo logico? 11 + 12

Quanti bit per indirizzo fisico? 9 + 12

Offset 12 = 2 + 10

$$2048 = 2^{11}$$

$$512 = 2^9$$

- Esercizio Memoria

Esercizi

Indirizzi logici a 32 bit, pagine da 8 KB, 1 Gb di memoria fisica

Quante entry per tabella di pagine di un solo livello?

Quanti entry per tabella pagine invertita?

- Esercizio Memoria

Esercizi

Indirizzi logici a 32 bit, pagine da 8 KB, 1 GB di memoria fisica

Quante entry per tabella di pagine di un solo livello?

$2^{13} = 8 \text{ KB}$, quindi 13 bit di offset, 19 bit per le pagine (32 -19), dunque la entry sono 2^{19}

Quanti entry per tabella pagine invertita?

1 GB sono 2^{30} byte, ogni pagina sono 2^{13} byte, quindi abbiamo 2^{17} entry ($2^{30} / 2^{13}$)

- Esercizio Memoria

Esercizi

Data una tabella delle pagine in memoria

- Se per un accesso in memoria ci vogliono 50 ns quanto ci vuole per un accesso alla memoria paginata?

- Se si aggiunge una TLB con hit rate di 75% con accesso ad essa pari a 2 nanosec qual è il tempo di accesso effettivo?

- Esercizio Memoria Virtuale

Esercizi

Data una tabella delle pagine in memoria

- Se per un accesso in memoria ci vogliono 50 ns quanto ci vuole per un accesso alla memoria paginata?

100 ns

- Se si aggiunge una TLB con hit rate di 75% con accesso ad essa pari a 2 ns
Qual è il tempo di accesso effettivo?

$$0.25 \times 100 + 2 + 0.75 \times 50 = 25 + 2 + 37,5 = 64,5 \text{ ns}$$

- Esercizio Memoria

Esercizi

Si consideri un sistema con memoria paginata ad un livello, la cui page table sia mantenuta in memoria principale. Il tempo di accesso alla memoria principale sia $t = 50ns$.

- (a) Qual è il tempo effettivo di accesso alla memoria?
- (b) Aggiungendo un TLB, con tempo di accesso $\epsilon = 1ns$, quale hit rate dobbiamo avere per un degrado delle prestazioni del 5% rispetto a t ?

- Esercizio Memoria

Esercizi

Si consideri un sistema con memoria paginata ad un livello, la cui page table sia mantenuta in memoria principale. Il tempo di accesso alla memoria principale sia $t = 50\text{ns}$.

- (a) Qual è il tempo effettivo di accesso alla memoria?
- (b) Aggiungendo un TLB, con tempo di accesso $\epsilon = 1\text{ns}$, quale hit rate dobbiamo avere per un degrado delle prestazioni del 5% rispetto a t ?

- (a) $EAT = 2t = 100\text{ns}$ (50ns per recuperare il frame number, 50ns per l'indirizzo reale)

- **Esercizio Memoria**

Esercizi

Si consideri un sistema con memoria paginata ad un livello, la cui page table sia mantenuta in memoria principale. Il tempo di accesso alla memoria principale sia $t = 50\text{ns}$.

- (a) Qual è il tempo effettivo di accesso alla memoria?
- (b) Aggiungendo un TLB, con tempo di accesso $\epsilon = 1\text{ns}$, quale hit rate dobbiamo avere per un degrado delle prestazioni del 5% rispetto a t ?

- (a) $EAT = 2t = 100\text{ns}$ (50ns per recuperare il frame number, 50ns per l'indirizzo reale)

- (b) Un degrado del 5% significa che $EAT = 1,05 * 50 = 52,5\text{ns}$. Ricordando che $EAT = \epsilon + \alpha t + (1 - \alpha)(2t)$, abbiamo che $EAT = \epsilon + 2t - \alpha t$, e quindi $\alpha = \frac{2t + \epsilon - EAT}{t} = \frac{100 + 1 - 52,5}{50} = 0,97$, ossia un hit rate del 97%.

- **Esercizio Memoria**

Esercizi

Si consideri un processo che generi la seguente stringa di riferimenti alle pagine virtuali:

0 1 2 0 1 4 5 0 2 3

- (a) Se il processo ha 3 frame, gestiti con LRU, quanti page fault vengono generati?

- Esercizio Memoria Virtuale

Esercizi

Si consideri un processo che generi la seguente stringa di riferimenti alle pagine virtuali:

0 1 2 0 1 4 5 0 2 3

- (a) Se il processo ha 3 frame, gestiti con LRU, quanti page fault vengono generati?

(a) Si generano 8 page fault:

	0	1	2	0	1	4	5	0	2	3
	0	1	2	0	1	4	5	0	2	
	0	1	2	0	1	4	5	0		
					2	0	1	4	5	
					2	2	1	4		
								1		
P	P	P		P	P	P	P	P	P	

- Esercizio Memoria Virtuale

Esercizi

Dato la seguente stringa di page reference

```
2,6,9,2,4,2,1,7,3,0,5,2,1,2,9,5,7,3,8,5
```

Quanti page fault occorrono, assumendo demand paging con 3 frame, con LRU, FIFO, OPT?

- Esercizio Memoria Virtuale

Esercizi

Dato la seguente stringa di page reference

```
2,6,9,2,4,2,1,7,3,0,5,2,1,2,9,5,7,3,8,5
```

Quanti page fault occorrono, assumendo demand paging con 3 frame, con LRU, FIFO, OPT?

LRU 17

FIFO 18

OPT 13

- Esercizio Memoria Virtuale

Esercizi

Dato la seguente stringa di page reference

0,6,3,0,2,6,3,5,2,4,1,3,0,6,1,4,2,3,5,7

Quanti page fault occorrono, assumendo demand paging con 3 frame, con LRU, FIFO, OPT?

LRU

FIFO

OPT

- Esercizio Memoria Virtuale

Esercizi

Dato la seguente stringa di page reference

0, 6, 3, 0, 2, 6, 3, 5, 2, 4, 1, 3, 0, 6, 1, 4, 2, 3, 5, 7

Quanti page fault occorrono, assumendo demand paging con 3 frame, con LRU, FIFO, OPT?

LRU 19

FIFO 16

OPT 13

- Esercizio Memoria Virtuale

Esercizi

Dato la seguente stringa di page reference

```
3,1,4,2,5,4,1,3,5,2,0,1,1,0,2,3,4,5,0,1
```

Quanti page fault occorrono, assumendo demand paging con 3 frame, con LRU, FIFO, OPT?

LRU 16

FIFO 15

OPT 11

- Esercizio Memoria Virtuale

Esercizi

- (a) Si descriva brevemente il concetto di *working set* $WS(t, \Delta)$, all'istante t con intervallo Δ .
- (b) Si consideri la seguente stringa di riferimenti (partendo con $t = 0$):

2 6 5 7 7 7 7 5 1 6 4

Cosa è $WS(10, 8)$, ossia dopo l'ultimo accesso?

- Esercizio Memoria Virtuale

Esercizi

- (a) Si descriva brevemente il concetto di *working set* $WS(t, \Delta)$, all'istante t con intervallo Δ .
- (b) Si consideri la seguente stringa di riferimenti (partendo con $t = 0$):

2 6 5 7 7 7 7 5 1 6 4

Cosa è $WS(10, 8)$, ossia dopo l'ultimo accesso?

- (a) Il working set è un'approssimazione della località del processo, ossia è l'insieme di pagine “attualmente” riferite. In generale $WS(t, \Delta)$ =insieme delle pagine riferite negli accessi $[(t - \Delta + 1), t]$.
- (b) $WS(10, 8) = \{1, 4, 5, 6, 7\}$

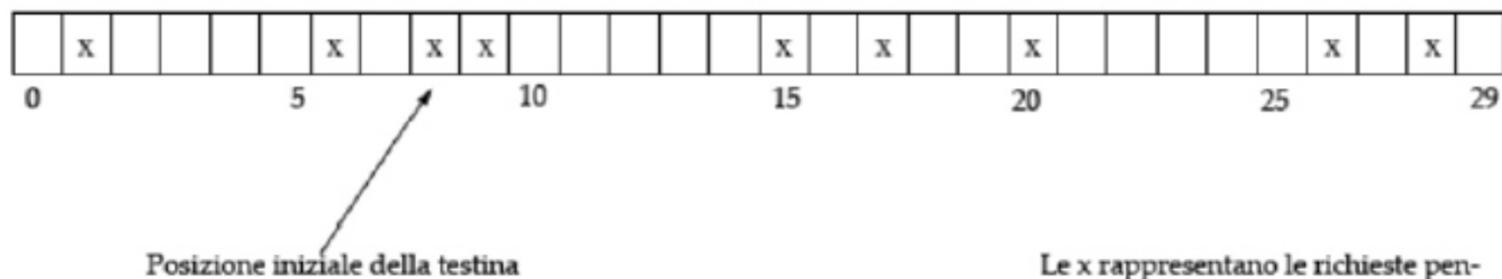
- **Esercizio Memoria Virtuale**

Argomenti Rilevanti

- Memoria di Massa
 - Concetti principali (HDDs e NVMe)
 - Mapping indirizzi e struttura disco
 - Schedulazione del disco
 - Algoritmi ed esercizi
 - **Gestione del Disco**
 - Connessioni a dispositivi di memorizzazione
 - Architetture RAID
 - Concetti, metodi

Esercizi

Nell'ambito della schedulazione delle richieste al disco, si consideri la situazione delle richieste pendenti e della testina illustrata in figura. Come si comporterà la testina nel caso in cui l'algoritmo di schedulazione utilizzato sarà l'SSF (Shortest Seek First)? Come si comporterà nel caso in cui sarà l'algoritmo dell'ascensore?



Le x rappresentano le richieste pendenti di accesso al disco

Esercizi

Algoritmo SSF (Shortest Seek First): risolte per prima le richieste più vicine alla posizione attuale della testina. La posizione successiva k è stabilita dalla formula seguente:

$$\arg \min_k |(P - P_k)|$$

Algoritmo dell'ascensore: risolte tutte le richieste in una direzione (salita, ad esempio), poi risolte quelle nell'altra direzione.

Garantisce la fairness del sistema, rendendo impossibile la starvation.

Algoritmo SSF soluzione:

8	9	6	1	15	17	20	26	28

Algoritmo dell'ascensore soluzione:

8	9	15	17	20	26	28	6	1

Esercizi

Assumendo 5000 cilindri e testina su 2150 (con richiesta precedente a 1805). Data la coda che segue

2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681

Cominciando dalla posizione corrente, qual è la distanza percorsa dalla testina con FCFS, SCAN, C-SCAN?

- Esercizio Memoria Virtuale

Esercizi

Assumendo 5000 cilindri e testina su 2150 (con richiesta precedente a 1805). Data la coda che segue

2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681

Cominciando dalla posizione corrente, qual è la distanza percorsa dalla testina con FCFS, SCAN, C-SCAN?

- Esercizio Memoria Virtuale

Esercizi

Assumendo 5000 cilindri e testina su 2150
(con richiesta precedente a 1805). Data la
coda che segue

2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681

FCFS, distanza totale 13011

SCAN

2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212,
544, 356 totale 7492

C-SCAN

2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523,
1618, 2069 totale 9917

- Esercizio Memoria Virtuale

Argomenti Rilevanti

- Sistemi I/O
 - Concetti principali
 - Polling/Interrupt
 - Direct Memory Access (DMA)
 - Interfacce per I/O
 - Caratteristiche dispositivi I/O
- **Sottosistema I/O del Kernel**
 - Strutture dati per gestione I/O

Esercizi

Descrivere l'accesso in memoria con DMA

- Esercizio I/O

Argomenti Rilevanti

- File System
 - Concetti principali
 - File, attributi, directory
 - Operazioni sui file
 - Chiamate di Sistema in Linux (esercizi)
 - Open, read, write, close, mmap
 - Comandi per la gestione di file e directory (esercizi shell linux)
 - Implementazione
 - Concetti principali
 - Struttura, Strutture dati
 - Directory
 - Metodi di allocazione dei file
 - Gestione dello spazio libero
 - Efficienza e prestazione
 - Gestione errori e recupero
 - Dettagli
 - Partizioni, montaggio, file system virtuale
 - File system distribuito

Esercizi

Si consideri un'implementazione di un filesystem con allocazione indicizzata con indice concatenato in cui:

- la dimensione di un blocco sia di 8 byte;
- un indirizzo richieda un byte per la sua memorizzazione.

Si calcoli l'indirizzo fisico corrispondente all'indirizzo logico 60 per un dato file.

- **Esercizio File System**

Esercizi

Si consideri un File System che utilizza inode con blocchi da 8 KB e puntatore a blocco di 4 byte

L'inode ha 12 puntatori diretti a blocchi, inoltre ha singoli, doppi e tripli riferimenti indiretti ai blocchi.

Quanto è il size massimo?

- Esercizio File System

Esercizi

Si consideri un File System che utilizza inode con blocchi da 8 KB e puntatore a blocco di 4 byte

L'inode ha 12 puntatori diretti a blocchi, inoltre ha singoli, doppi e tripli riferimenti indiretti ai blocchi.

Quanto è il size massimo?

$$12 \times 8 \text{ KB} + 2048 \times 8 \text{ KB} + 2048 \times 2048 \times 8 \text{ KB} + 2048 \times 2048 \times 2048 \times 8 \text{ KB} = 64 \text{ terabyte}$$

- Esercizio File System

Argomenti Rilevanti

- Linux
 - Overview
 - Gestione di processi e thread (task)
 - Scheduling
 - Gestione della memoria
 - Memoria virtuale
 - Paging, page replacement, free memory
 - Caricamento ed esecuzione
 - **File system (ext)**
 - Sistema I/O