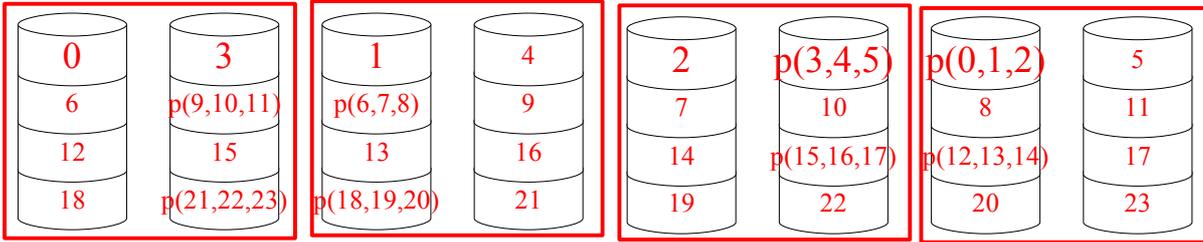


Sistemi Operativi — A.A. 2007-2008, prova scritta del 10 aprile 2008.

Considera 8 dischi in raid 05 (raid0 a gruppi di 2). Mostra come sono disposti i blocchi logici nei blocchi fisici compilando il seguente schema (indica con P la parità).



Spiega il **write-starving-reads problem** nell'ambito dello scheduling del disco e descrivi la tecnica usata dallo scheduling “**anticipatory**” per affrontare tale problema.

vedi materiale didattico

3. Considera la politica di scheduling round robin. Supponi che in un sistema ci siano N processi interattivi tutti con lo stesso comportamento. Ciascuna interazione dà luogo ad un cpu burst che richiede la cpu per un tempo c. Il quanto dura un tempo q.

Se $c < q$ quanto tempo aspetta al più un **processo** in coda ready prima di ottenere la cpu?

Un processo aspetta $(N-1)c$ per ottenere la cpu.

Se $c < q$ quanto tempo aspetta al più l'**utente** prima che la cpu finisca di elaborare l'interazione?

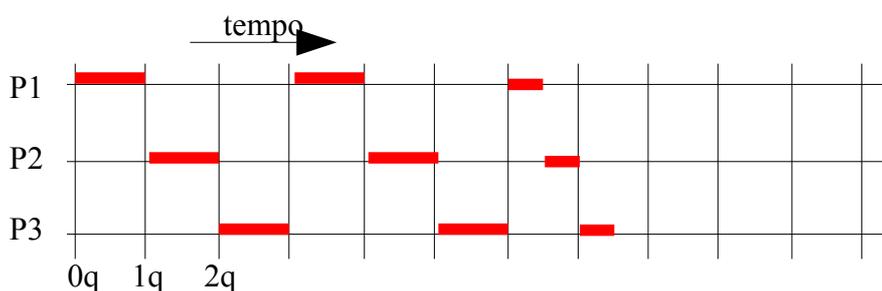
L'utente aspetta $(N-1)c + c = Nc$

Se $c > q$ quanto tempo aspetta al più un **processo** in coda ready prima di ottenere la cpu?

Un processo aspetta al più $(N-1)q$ per ottenere la cpu.

Se $c > q$ quanto tempo aspetta al più l'**utente** prima che la cpu finisca di elaborare l'interazione? Giustifica la risposta mostrando uno schema di allocazione della CPU per $c=2.5q$ e $N=3$.

Evidenzia come la cpu si alterna tra i processi nel tempo

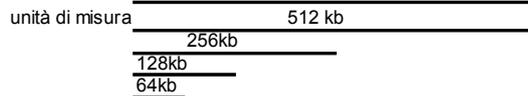
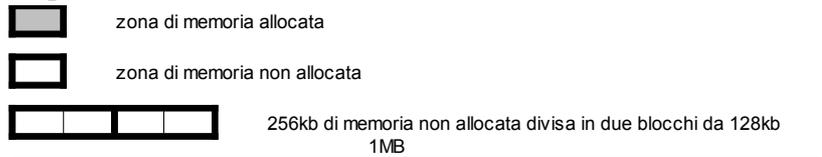


Quanto tempo attende l'utente in generale?

Sistemi Operativi — A.A. 2007-2008, prova scritta del 10 aprile 2008.

Poiché $c > q$, esistono a e b tali che $c = a \cdot q + b$. Per svolgere la sua interazione ciascun processo ha bisogno di a quanti interi più un ultimo quanto parziale di durata b . Il caso peggiore si ha quando arriva contemporaneamente una interazione per ciascun processo. Si avrà un certo ordine per la coda ready. L'ultimo processo nella coda attende prima dell'esecuzione di ogni quanto un tempo $(N-1)q$, tranne l'ultima volta che attende $(N-1)b$. L'attesa totale dell'utente è $(N-1)qa + (N-1)b + c = (N-1)c + c = Nc$

4. Considera una zona di memoria di 1 megabyte gestita tramite un buddy system. Qui sotto ti vengono proposte delle situazioni descritte con uno stato del buddy system, una operazione da effettuare (allocazione o deallocazione) e la politica di coalescing: “immediata” o “lazy” (cioè coalescing solo quando strettamente necessario). Mostra come lo spazio di memoria è suddiviso dopo ogni operazione, quali blocchi sono allocati e indica anche quanti splitting o quanti coalescing sono stati effettuati, **se l'operazione è possibile**. Eventualmente **indica se l'operazione è impossibile**.



operazione	prima	numero di splitting:	numero di coalescing:	dopo
1 alloc 64 kb coalescing immediato		4	0	
2 free A coalescing immediato		0	4	
3 alloc 64 kb coalescing lazy		4	0	
4 free A coalescing lazy		0	0	
5 alloc 512 kb coalescing lazy		0	3	
6 free A coalescing immediato		0	1	
7 free A coalescing immediato		0	0	
8 alloca 256kb coalescing immediato				
9 alloca 256kb coalescing lazy				

5. Dai la definizione di Working Set per un processo specificando il significato del parametro delta Δ .

vedi materiale didattico

Confronta la politica “resident set=working set” con la politica LRU. Analogie e differenze.

WS può essere considerata una versione di LRU con taglia del resident set variabile dinamicamente in base alle necessità del processo.

Supponi che ci sia un processo che abbia due fasi di esecuzione A e B, consecutive. A e B sono molto locali (le pagine accedute non variano all'interno di una fase) ma accedono a pagine distinte. Che comportamento ti aspetti dal working set durante le due fasi? e nel transitorio da A a B?

supponendo che delta sia abbastanza grande, durante A, a regime, il WS conterrà le pagine necessarie alla sua esecuzione, come pure durante B.

Nel transitorio conterrà pagine sia di A che di B e quindi aumenta la sua dimensione.