



Embedded Systems

I Sistemi Operativi per Applicazioni Multimediali

Docente:

William Fornaciari

Politecnico di Milano

fornacia@elet.polimi.it



Sommario

- Introduzione
- *File* Multimediali
- Sistemi Operativi per Applicazioni Multimediali
 - ▶ *Process Scheduling* per Applicazioni Multimediali
 - ▶ *File System* per Applicazioni Multimediali
 - ▶ *Disk Scheduling* per Applicazioni Multimediali



Introduzione

- Multimedia
- Applicazioni Multimediali
- Video on Demand



Introduzione

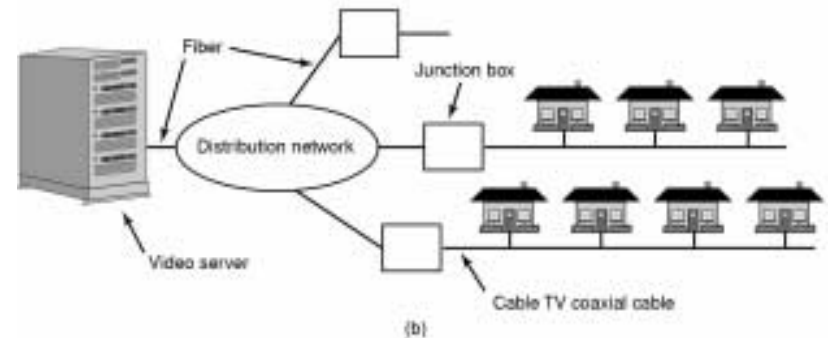
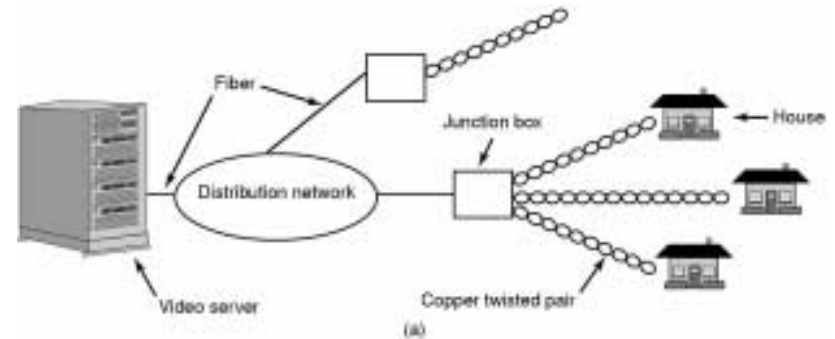
- Multimedia
 - ▶ Più di un *media*
- Applicazioni Multimediali
 - ▶ Applicazioni che trattano dati relativi ad almeno due *media continui* che devono essere riprodotti in un certo intervallo di tempo
 - ▶ Il crescente numero di applicazioni che trattano film digitali, video clip e musica impone la necessità di sistemi operativi pensati appositamente per gestirle
 - Riproduzione di CD-ROM/DVD
 - *Download/streaming* di video
 - *Video Editing*
 - Videogiochi



Introduzione

- *Video on Demand*

- ▶ *Video Server*
- ▶ *Rete di Distribuzione*
- ▶ *ADSL (a)*
 - Canale dedicato
 - *Bandwidth* garantita
 - *Bandwidth* stretta
- ▶ *Cable-TV (b)*
 - Canale condiviso
 - *Bandwidth* non garantita
 - *Bandwidth* larga
- ▶ *Set-Top Box/PC*





Introduzione

- Le applicazioni multimediali sono caratterizzate da
 - ▶ *Data rate* estremamente elevati
 - Occhi ed orecchie processano un'elevata quantità di informazione
 - ▶ Necessità di riproduzione in *real-time*
 - Le immagini devono essere fornite con cadenza precisa e predefinita (NTSC, PAL, SECAM): 25-30 *frame/s*
 - I suoni devono essere forniti con *rate* a varianza strettamente limitata per evitare *jitter* ai quali l'orecchio è molto sensibile
 - Le caratteristiche *real-time* necessarie vengono spesso descritte con i parametri relativi alla **qualità di servizio**
 - Si ottiene riservando in anticipo le risorse (%CPU, %memoria, %*network bandwidth*) e decidendo di volta in volta se ammettere nuovi utenti (***Admission Control Algorithm***)



Introduzione

- *Data rate di alcuni dispositivi*
 - *Multimedia I/O devices*
 - *High-Performance I/O devices*

| Source | Mbps | GB/hr |
|---------------------------------------|-------|-------|
| Telephone (PCM) | 0.064 | 0.03 |
| MP3 music | 0.14 | 0.06 |
| Audio CD | 1.4 | 0.62 |
| MPEG-2 movie (640 \times 480) | 4 | 1.76 |
| Digital camcorder (720 \times 480) | 25 | 11 |
| Uncompressed TV (640 \times 480) | 221 | 97 |
| Uncompressed HDTV (1280 \times 720) | 648 | 288 |
| Fast Ethernet | 100 | |
| EIDE disk | 133 | |
| ATM OC-3 network | 156 | |
| SCSI UltraWide disk | 320 | |
| IEEE 1394 (FireWire) | 400 | |
| Gigabit Ethernet | 1000 | |
| SCSI Ultra-160 disk | 1280 | |



File Multimediali

- Generalità
- Audio
 - ▶ Codifica
 - ▶ Compressione
- Video
 - ▶ Codifica
 - ▶ Compressione

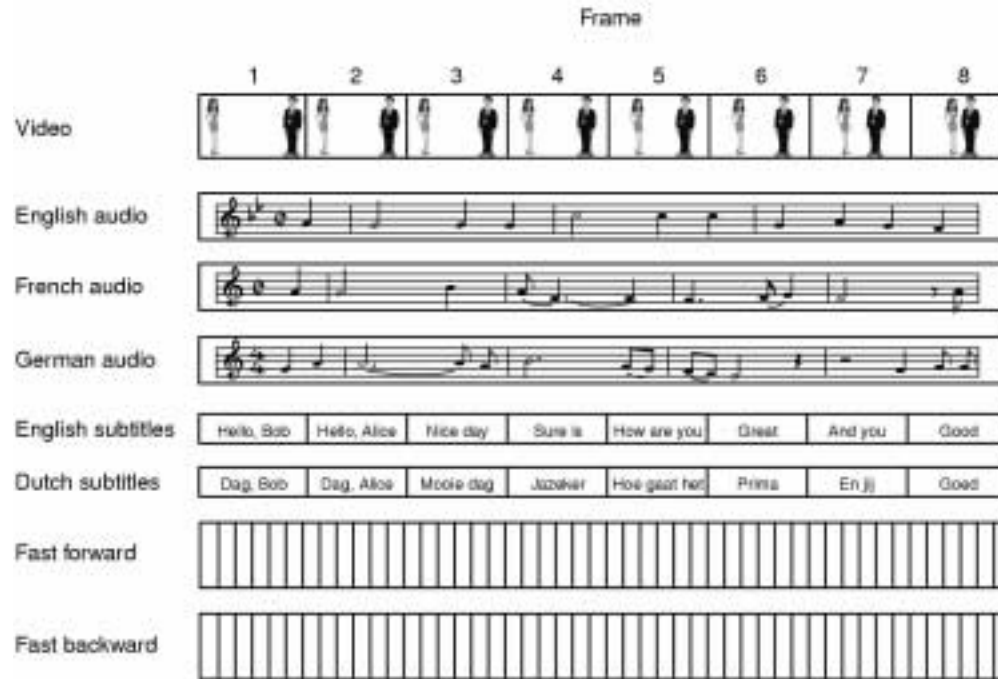


File Multimediali

- I sistemi operativi classici sono stati pensati per trattare *file* di testo
 - Una sequenza lineare di *byte* senza particolari strutture interne
- Video e audio oltre ad essere diversi dal testo sono molto diversi anche tra di loro
 - Dispositivi di acquisizione
 - Struttura interna
 - Dispositivi di riproduzione
- Inoltre in un film digitale
 - Esiste la possibilità di avere più tracce audio (ed eventuali sottotitoli) in varie lingue



File Multimediali



- Il *file system* deve tenere traccia di *file* multipli che devono restare sincronizzati



File Multimediali

- Audio

- ▶ Codifica Audio Digitale

- Un'onda sonora è un'onda acustica monodimensionale che viene trasformata da un microfono in un segnale elettrico analogico che varia in funzione del tempo
 - Questo viene campionato ad un'opportuna frequenza e convertito in digitale introducendo un certo errore di quantizzazione
 - *Pulse Code Modulation* (7/8 bit), CD-Audio (16 bit)

- ▶ Compressione Audio Digitale

- E' basata su alcuni modelli psicoacustici che descrivono il comportamento dell'orecchio umano
 - MPEG Layer 3 (MP3): compressione fino a 10x



File Multimediali

- Video

- ▶ Codifica Video Analogica

- E' meno problematica rispetto all'audio perché sfrutta la persistenza dell'immagine sulla della retina: bastano 50 immagini/s per rendere l'illusione della continuità delle immagini
 - *Scan* e CRT
 - Dato che gli standard (NTSC, PAL, SECAM) prevedono solo 25-30 *frame/s* (per problemi di banda) si usa l'*interlacing*
 - Si riproducono alternativamente le linee pari e quelle dispari

- ▶ Codifica Video Digitale

- Semplice sequenza di frame
 - I monitor digitali usano la RAM Video per fare il *rescan* e raggiungere anche gli 80 *frame/s*
 - Non serve l'*interlacing*



File Multimediali

- Compressione Video

- ▶ E' fondamentale per ridurre la banda necessaria
- ▶ Gli algoritmi di compressione possono essere asimmetrici e *lossy*
- ▶ Gli standard
 - JPEG (*Joint Photographic Experts Group*)
 - Compressione di singole immagini
 - MPEG (*Motion Pictur Experts Group*)
 - Sfrutta la ridondanze
 - » Spaziale: ogni immagine è compressa con JPEG
 - » Temporale: si tiene traccia solo dei cambiamenti tra un'immagine e l'altra
 - Tre tipi di *frame*: *I-frame*, *P-frame* e *B-frame*



Sistemi Operativi per Applicazioni Multimediali

- I Sistemi Operativi che devono supportare applicazioni multimediali differiscono da quelli classici per tre aspetti principali:
 - Lo *Scheduling* dei Processi (*Process Scheduling*)
 - Il *File System*
 - Lo *Scheduling* del Disco (*Disk Scheduling*)



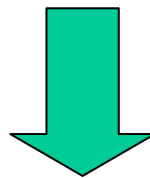
Process Scheduling per Applicazioni Multimediali

- *Scheduling* di Processi Omogenei
 - *Round-Robin* Temporizzato
- *Real-Time Scheduling*
 - *Rate Monotonic Scheduling*
 - *Earliest Deadline First Scheduling*



Process Scheduling per Applicazioni Multimediali

- *Scheduling* di Processi Omogenei
 - ▶ Video Server con un numero fisso di film con le stesse caratteristiche
 - *Frame rate, video resolution, data rate, etc.*
 - ▶ Si ha un processo per ogni film (o *stream*)
 - Ogni processo legge un *frame* e lo invia all'utente
 - Tutti i processi hanno la stessa importanza

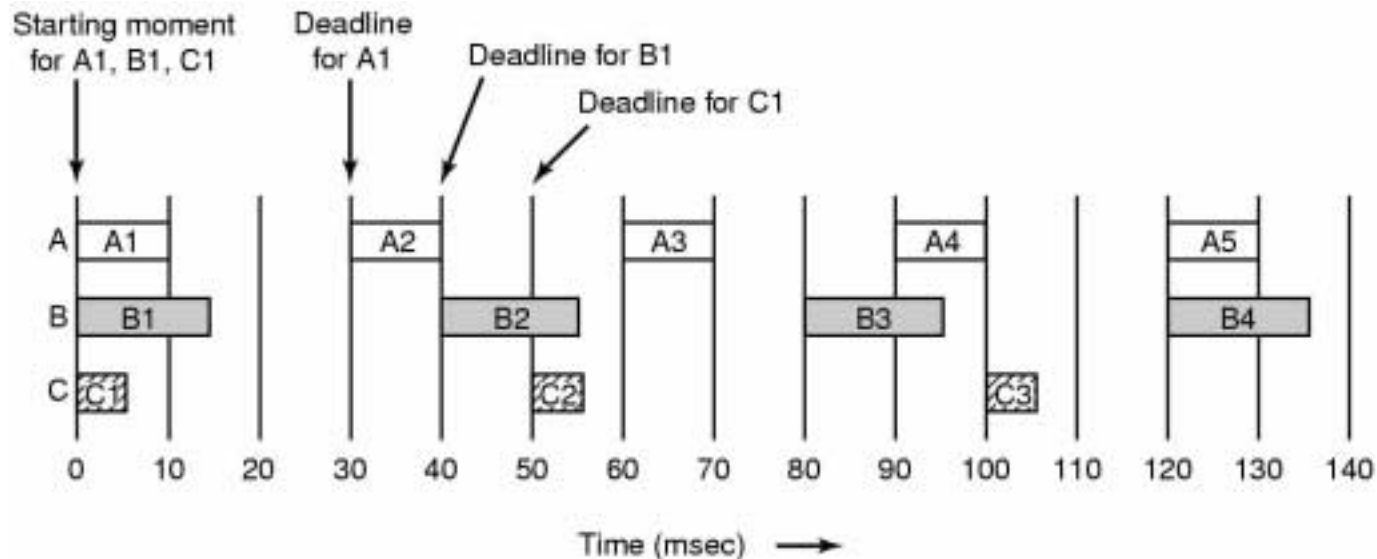


Round-Robin Temporizzato



Process Scheduling per Applicazioni Multimediali

- Ma in realtà le cose non stanno così!
 - ▶ Molti parametri variano (anche dinamicamente)
 - Numero di utenti
 - *Frame size* e *Frame rate*
 - Periodi dei processi, carico di lavoro e *deadline*





Process Scheduling per Applicazioni Multimediali

- Se il sistema conosce la frequenza, il carico e le deadline dei processi si possono adottare algoritmi di *Real-time Scheduling*
 - Si possono avere anche più processi per *stream*
 - Es. Audio e Video
 - Deve esistere uno scheduling fattibile:

$$\sum_{i=1..m} C_i / P_i \leq 1$$

- Si usano algoritmi *preemptive* statici o dinamici
 - *Rate Monotonic Scheduling* (statico)
 - *Earliest Deadline First Scheduling* (dinamico)



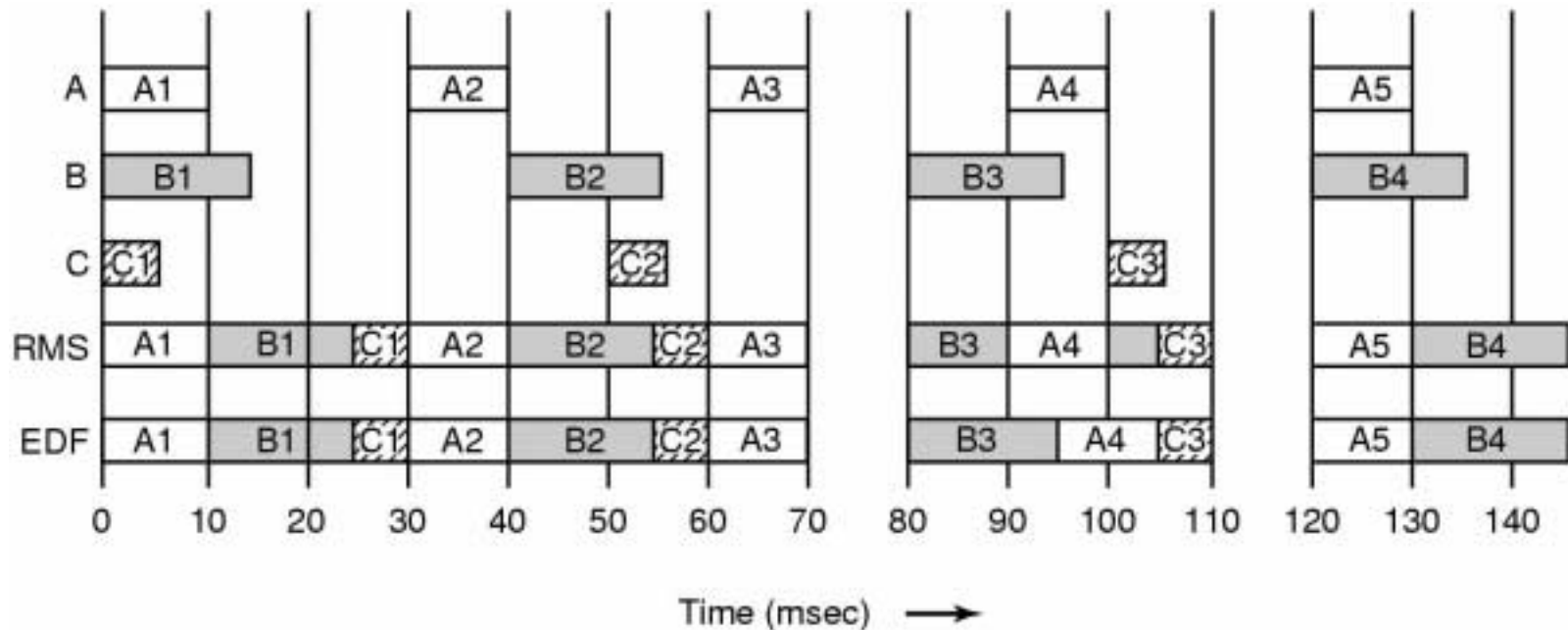
Process Scheduling per Applicazioni Multimediali

- *Rate Monotonic Scheduling*
 - ▶ 5 Condizioni iniziali
 - ▶ Assegna ad ogni processo una priorità fissa pari alla frequenza con cui dovrebbe essere eseguito
 - ▶ Lo *scheduler* esegue sempre il processo pronto a più alta priorità (eventualmente fa *preemption*)
- *Earliest Deadline First Scheduling*
 - ▶ Si basa su meno ipotesi: ogni processo annuncia la sua presenza e dichiara la sua *deadline*
 - ▶ Usa una lista di processi ordinati per *deadline*
 - ▶ Lo *scheduler* esegue (facendo eventualmente *preemption*) il processo pronto più prossimo alla scadenza della sua *deadline* (il primo della lista)



Process Scheduling per Applicazioni Multimediali

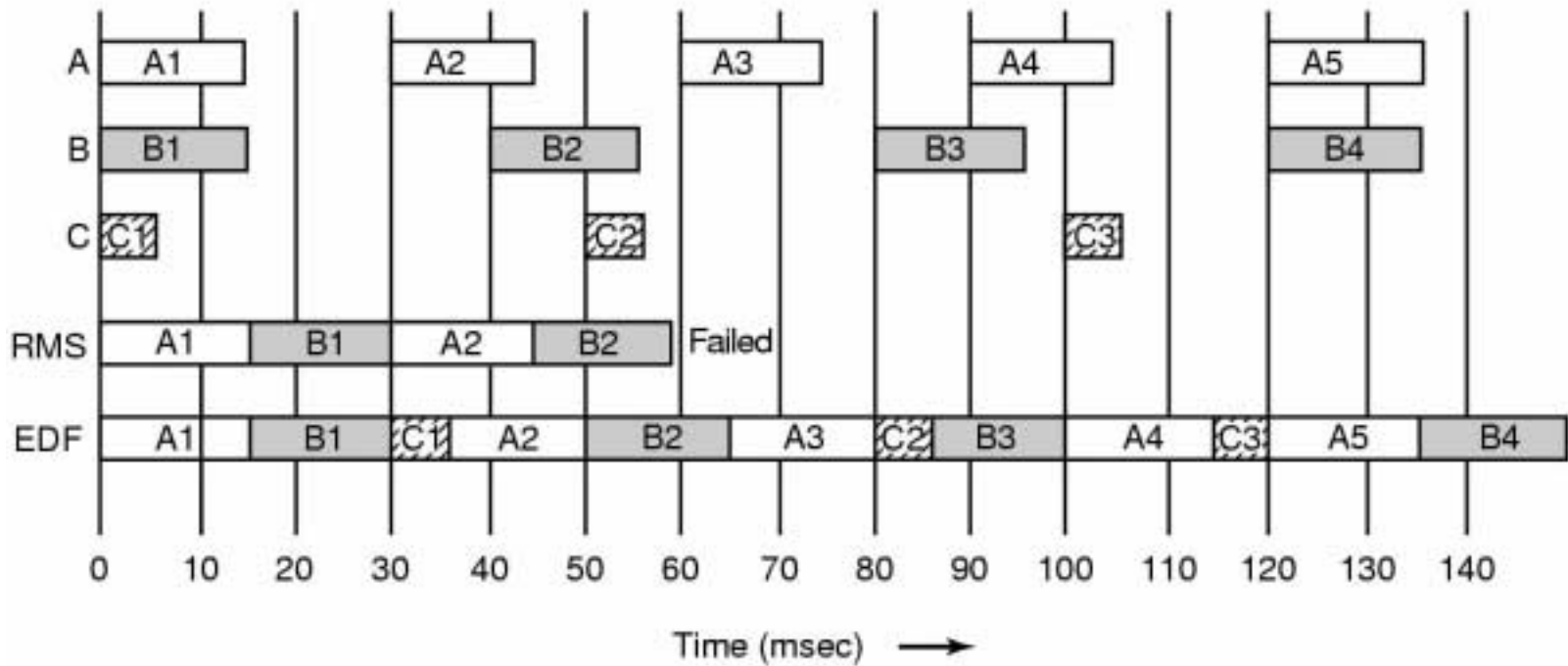
- Esempio in cui i due algoritmi danno gli stessi risultati





Process Scheduling per Applicazioni Multimediali

- Esempio in cui i due algoritmi danno risultati diversi





Process Scheduling per Applicazioni Multimediali

- In realtà RMS può essere usato con sicurezza solo se

$$\sum_{i=1..m} C_i / P_i \leq m(2^{1/m} - 1)$$

- EDF invece è sicuro per ogni insieme di processi che sia schedulabile ma è un algoritmo molto più complesso
- In pratica in un *Video Server* reale se ci si trova entro il limite di certezza si usa RMS altrimenti EDF



File System per Applicazioni Multimediali

- *File System* tradizionali
- Funzionalità VCR (*Video Cassette Recorder*)
- *Near Video on Demand*
- Posizionamento dei file (*File Placement*)
- *Caching* dei *file*



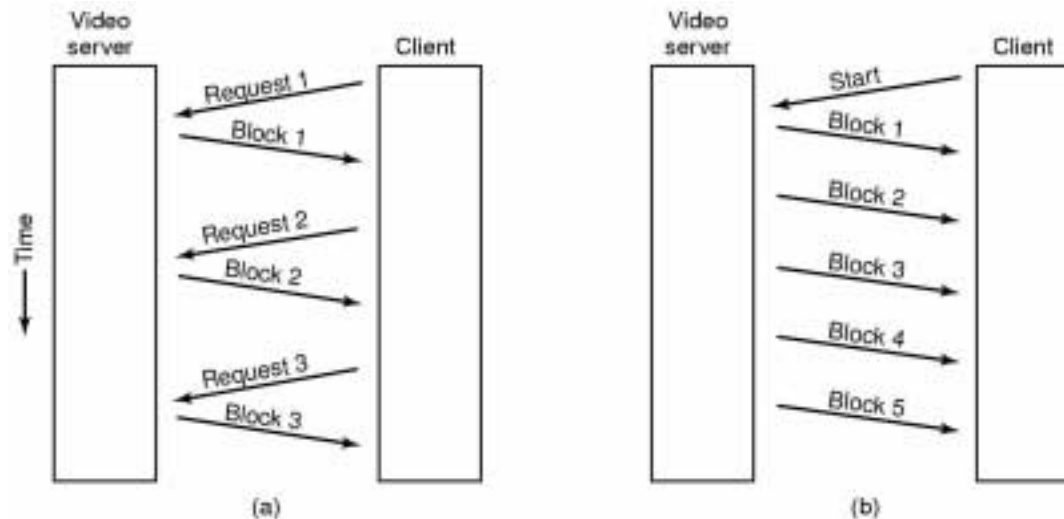
File System per Applicazioni Multimediali

- *File system* tradizionali
 - ▶ I processi fanno richiesta di dati tramite le *system call*
 - ▶ Nelle applicazioni multimediali, considerando i requisiti di tempo reale, i processi dovrebbero fare richieste temporizzate con estrema precisione e riuscire ad ottenere i dati con ritardi nulli da parte del *Video Server*
 - ▶ Tutto ciò è molto difficile da ottenere soprattutto se le richieste non sono pianificate e le opportune risorse non sono riservate in anticipo
- E' necessario un paradigma diverso



File System per Applicazioni Multimediali

- Funzionalità VCR (*Video Cassette Recorder*)
 - ▶ Un processo effettua un'opportuna *system call* ed il server comincia ad inviare i dati (*frame*) con la cadenza opportuna fino a che non viene fermato
 - E' compito del *client* riuscire a gestire i *frame*!
 - ▶ *Push Server* (a) vs. *Pull Server* (b)





File System per Applicazioni Multimediali

- Funzioni di controllo del VCR
 - ▶ Pausa
 - Una system call blocca il flusso ed un'altra lo fa ripartire dal punto in cui si era interrotto
 - Basta tenere traccia del numero del *frame* dal quale ripartire
 - Problema: *performance* vs. spreco di risorse
 - ▶ *Rewind*
 - Il flusso riparte dal *frame* 0



File System per Applicazioni Multimediali

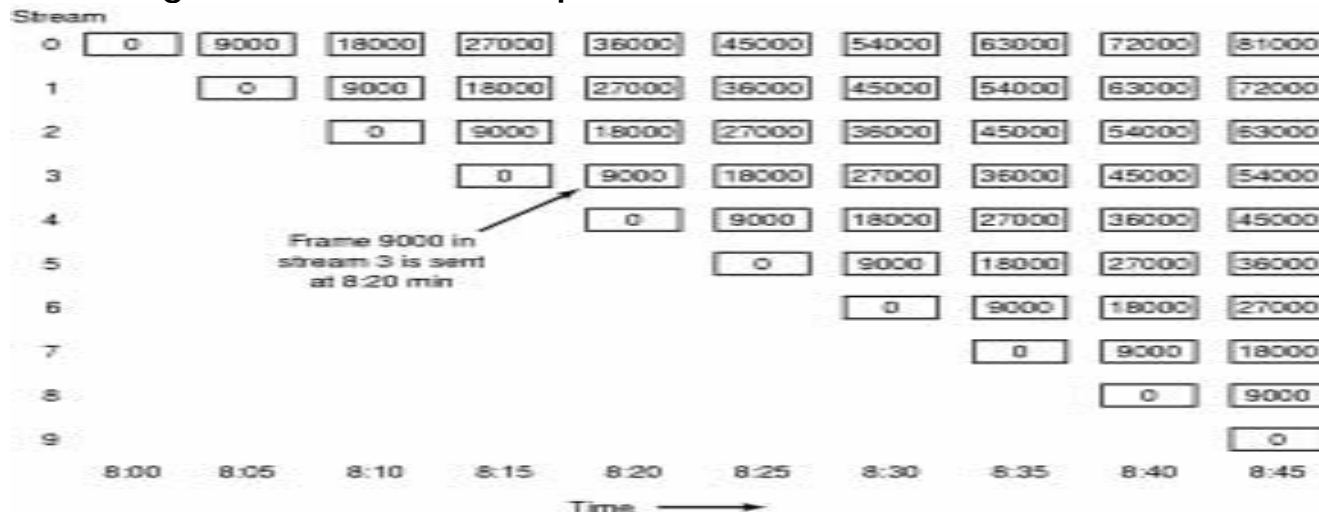
- Funzioni di controllo del VCR
 - ▶ *Fast forward/backward*
 - Sono operazioni più problematiche
 - Senza compressione: basta saltare *k frame* alla volta
 - Con compressione MPEG
 - Non è possibile saltare perchè servono gli *I-frame*
 - Non è fattibile aumentare il *frame rate*
 - Sono necessari file speciali preparati in precedenza contenenti i *frame* opportuni e compressi a loro volta in MPEG
 - Per effettuare le letture veloci si usano questi file
 - Ciò richiede spazio extra su disco e comunque il fattore di accelerazione è imposto dalla struttura dei *file* speciali



File System per Applicazioni Multimediali

- *Near Video on Demand*

- ▶ k utenti che guardano lo stesso film impongono un carico uguale a quello di k utenti che guardano film diversi
- ▶ E' possibile adottare un trucco per migliorare la situazione
 - I film non iniziano più in momenti arbitrari ma, ad esempio, ogni 5 minuti: per un film di 2 ore servono solo 24 stream che vengono condivisi tra più utenti





File System per Applicazioni Multimediali

- *Near Video on Demand*
 - ▶ Non ci sono più le funzionalità VCR ma un utente può passare ad un altro *stream* (in anticipo o in ritardo)
 - ▶ Ottimizzazione basata sulle richieste
 - I film partono ogni n minuti solo se c'è stata almeno una richiesta nel frattempo: in questo modo si eliminano gli *stream* inutili
 - Non è più possibile con certezza cambiare *stream* per andare avanti o indietro: lo *stream* necessario potrebbe non esserci!
 - ▶ Le funzionalità VCR possono essere introdotte solo dotando i *client* di opportuni *buffer* e della capacità di leggere più *stream* contemporaneamente
 - Gli *spostamenti* avvengono facendo riferimento ai dati locali
 - Problemi in più si hanno se si va oltre i dati *bufferizzati*



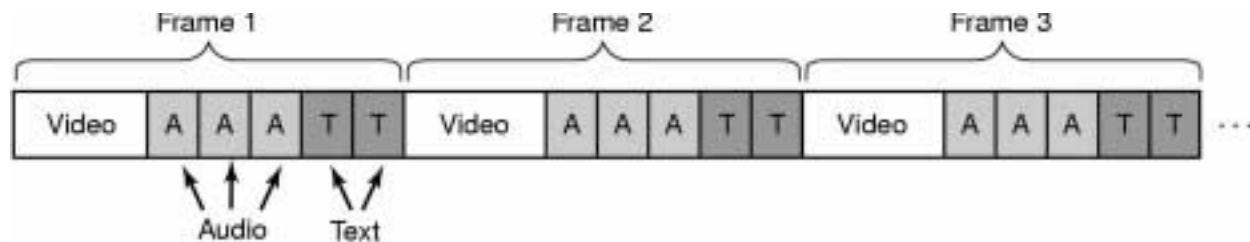
File System per Applicazioni Multimediali

- Posizionamento dei *file*
 - ▶ I *file* multimediali
 - Sono molto grandi
 - Sono spesso scritti una volta sola ma letti molte volte
 - Tendono ad essere acceduti sequenzialmente
 - ▶ La riproduzione di questi *file* deve soddisfare dei criteri relativi alla qualità di servizio
 - ▶ E' quindi necessario adottare particolari strategie di posizionamento



File System per Applicazioni Multimediali

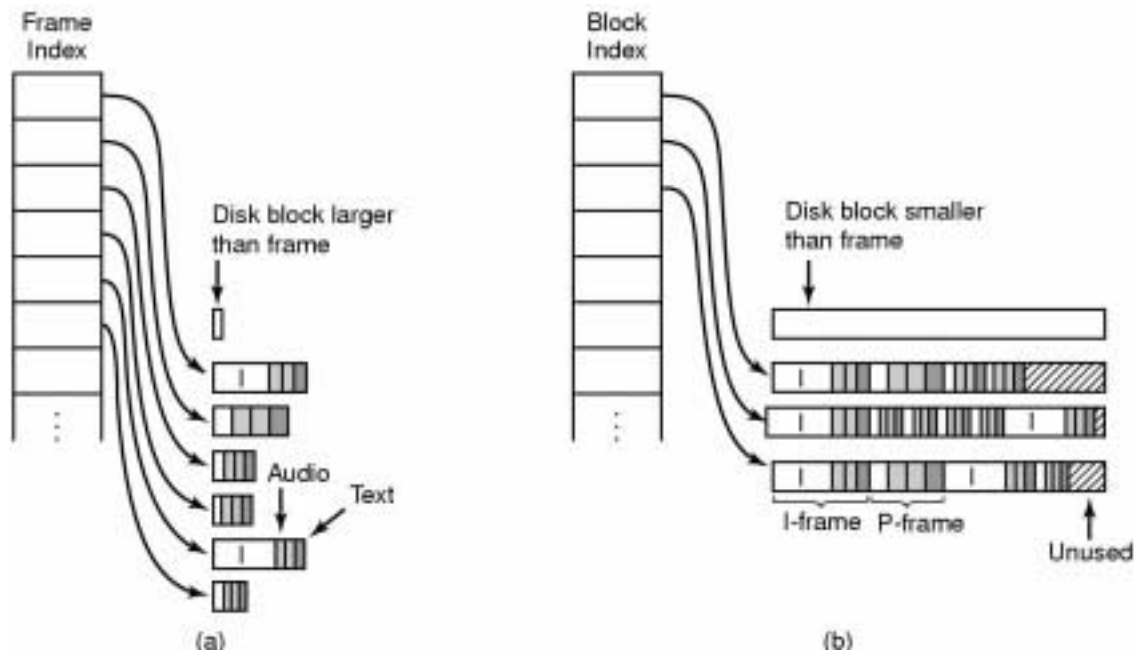
- Posizionamento di un *file* su un disco singolo
 - ▶ I *file* devono essere riprodotti alla velocità richiesta e senza *jitter*, evitando quindi *seek* multipli per accedere ad un frame: **allocazione contigua dei blocchi di un *file***
 - ▶ Bisogna però gestire più file correlati (video, audio, testo)
 - **Interleaving:** video, audio, e testo in un unico *file* contiguo
 - Si legge l'intero blocco e si usa quel che serve
 - FF/FB necessitano di apposite strutture dati
 - Approccio utile solo su sistemi *single output stream*





File System per Applicazioni Multimediali

- Posizionamento di un *file* su un disco singolo
 - Strategie di allocazione non contigue
 - Small Block Model (a): piccoli blocchi contigui per *frame*
 - Large Block Model (b): più *frame* in un solo grande blocco





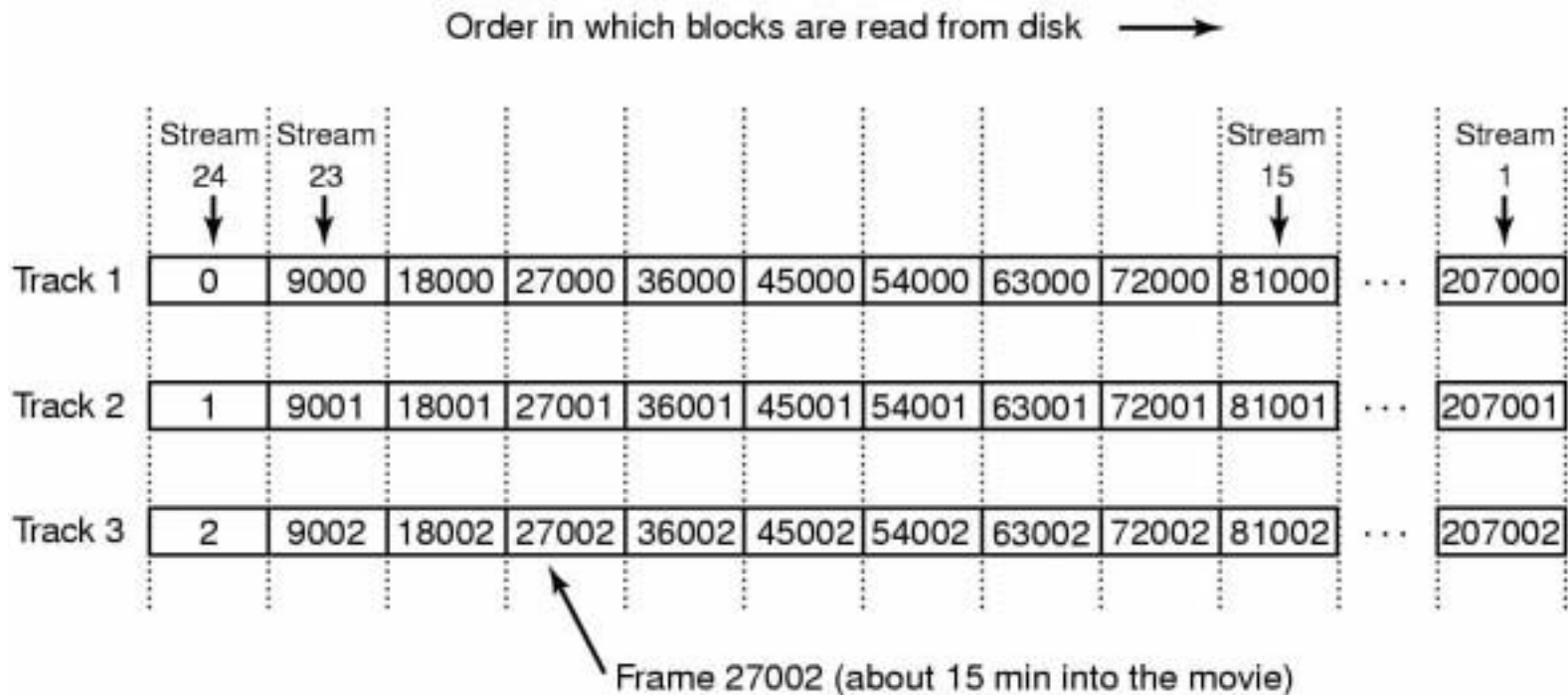
File System per Applicazioni Multimediali

- Trade-off tra SBM e LBM
 - ▶ SBM
 - Frame index
 - Pesante utilizzo della RAM di sistema durante la riproduzione
 - Poco spreco di spazio su disco
 - ▶ LBM
 - Block Index (senza *splitting*)
 - Basso utilizzo di RAM di sistema durante la riproduzione
 - Notevole spreco di spazio su disco (frammentazione interna)
 - Block Index (con *splitting*)
 - Basso utilizzo di RAM di sistema durante la riproduzione
 - Poco spreco di spazio su disco
 - Necessità di un maggior numero di operazioni di seek



File System per Applicazioni Multimediali

- Posizionamento dei file per il *Near Video on Demand*
 - Si tengono i frame *correlati* (di vari *stream*) sulla stessa traccia e si leggono tutti con un solo *seek*





File System per Applicazioni Multimediali

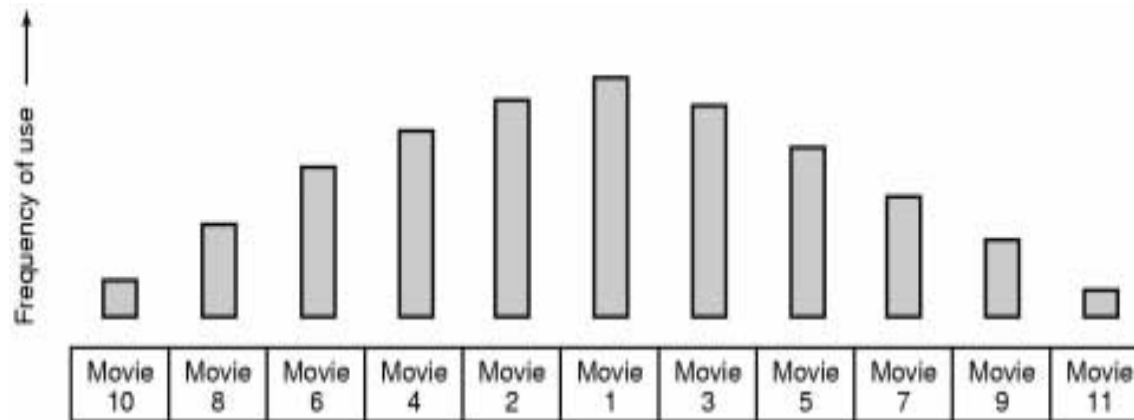
- Posizionamento di più file su un disco singolo
 - ▶ In un *Video Server* ci saranno molti film: posizionarli casualmente sul disco causa numerosi spostamenti delle testine quando più film sono visti contemporaneamente
 - Bisogna posizionare i film basandosi sulla loro popolarità
 - ▶ Legge di Zipf
 - Classificando i film (libri, pagine web, parole, etc.) in base alla loro popolarità, la probabilità che il prossimo utente scelga il film in posizione k è C/k , dove C è una costante di normalizzazione tale che

$$\sum_{k=1..n} C/k = 1$$



File System per Applicazioni Multimediali

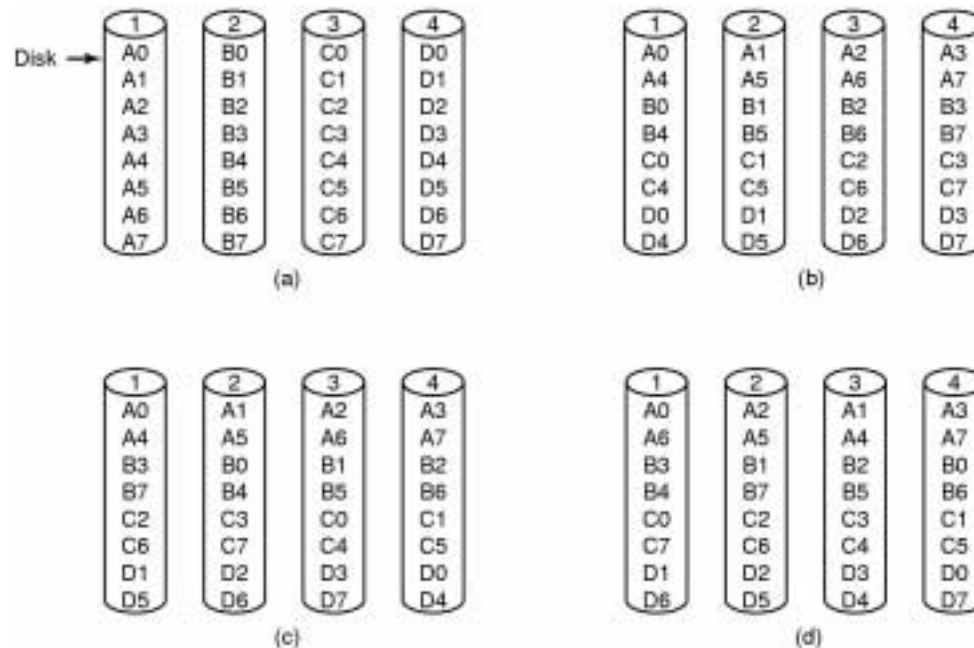
- Posizionamento di più file su un disco singolo
 - ▶ La legge di Zipf ci dice che il film più popolare è visto il doppio rispetto al secondo, il triplo rispetto al terzo, etc.
 - ▶ Basandosi su questa legge è stato messo a punto un algoritmo, chiamato *organ-pipe algorithm*, che consiste nel posizionare i film più popolari al centro del disco minimizzando così gli spostamenti delle testine





File System per Applicazioni Multimediali

- Posizionamento di più file su dischi multipli
 - ▶ Per ottenere maggiori prestazioni ed offrire più film i *Video Server* utilizzano più dischi paralleli (*disk farm*)
 - ▶ Possibili configurazioni





File System per Applicazioni Multimediali

- Posizionamento di più file su dischi multipli
 - Un file per disco (*a*)
 - Spreco, poca tolleranza ai guasti, carico sbilanciato
 - Suddivisione dei film su più dischi, con ordine simile (*b*)
 - Carico sbilanciato
 - Suddivisione dei film su più dischi, con ordine sfasato (*c*)
 - Suddivisione dei film su più dischi, casualmente (*d*)
- ▶ Quanti dischi utilizzare per ogni film?
 - *Wide stripping*: un film su molti dischi
 - Carico bilanciato ma molto sensibile ai guasti
 - *Narrow stripping*: un film su pochi dischi
 - Rischio di carico sbilanciato ma i guasti provocano meno danni



File System per Applicazioni Multimediali

- *Caching dei file*
 - ▶ Le classiche politiche di *caching* dei *file* non sono molto adatte per i *file* multimediali (es. LRU) perché in generale non valgono i classici principi di località/spazialità
 - ▶ Tuttavia, facendo alcune considerazioni è possibile sfruttare ancora i vantaggi della *cache*
 - ▶ *Block Caching*
 - Se un film ha più spettatori temporalmente vicini i blocchi letti possono essere messi in *cache* e riutilizzati con efficienza
 - ▶ *File Caching*
 - Necessario se non c'è abbastanza spazio su disco e i film vanno letti da DVD o nastro
 - Si tengono su disco i film più richiesti e i primi minuti degli altri (che verranno caricati all'occorrenza durante la riproduzione dei primi minuti)



Disk Scheduling per Applicazioni Multimediali

- Generalità
- *Disk Scheduling* Statico
- *Disk Scheduling* Dinamico
 - *Scan-EDF*



Disk Scheduling per Applicazioni Multimediali

- Le applicazioni multimediali necessitano di elevati *data rate* e di rilascio di dati in tempo reale
- Nel caso di *Video Server* è necessario anche massimizzare il numero di utenti contemporanei
- Tutto ciò impone l'adozione di algoritmi particolari anche per lo scheduling del disco
 - Algoritmi Statici
 - Algoritmi Dinamici



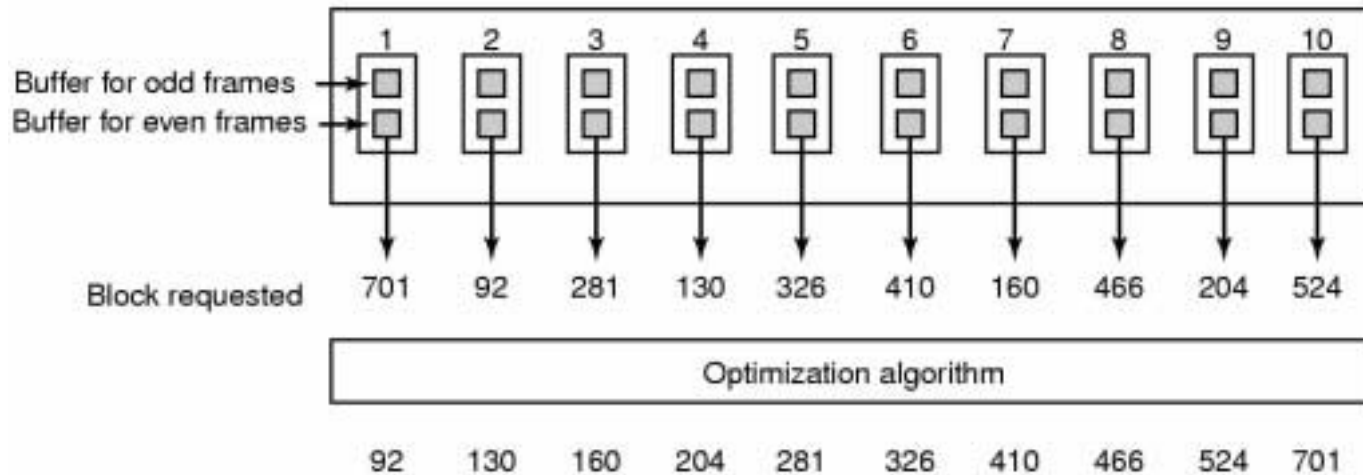
Disk Scheduling per Applicazioni Multimediali

- *Disk Scheduling* Statico
 - ▶ Obiettivo: sfruttare la predicibilità di *stream* simili
 - Ogni stream attivo impone al sistema un carico noto
 - ▶ Lo *scheduler* si basa sul concetto di *round*
 - Supponendo *stream* tutti simili (frame rate, risoluzione, etc.) ogni *round* ha una durata pari al *frame time*
 - Le richieste arrivano contemporaneamente ed è quindi possibile ottimizzare l'accesso al disco affinché sia possibile soddisfarle tutte entro il *round*
 - Ad esempio soddisfacendole in base al numero di cilindro (algoritmo dell'ascensore) minimizzando quindi il tempo di seek



Disk Scheduling per Applicazioni Multimediali

- *Disk Scheduling* Statico



- ▶ Se si finisce in anticipo rispetto al *round* vuol dire che è possibile servire più utenti contemporaneamente
- ▶ Per mantenere un elevato data rate è necessario usare un *double buffering*



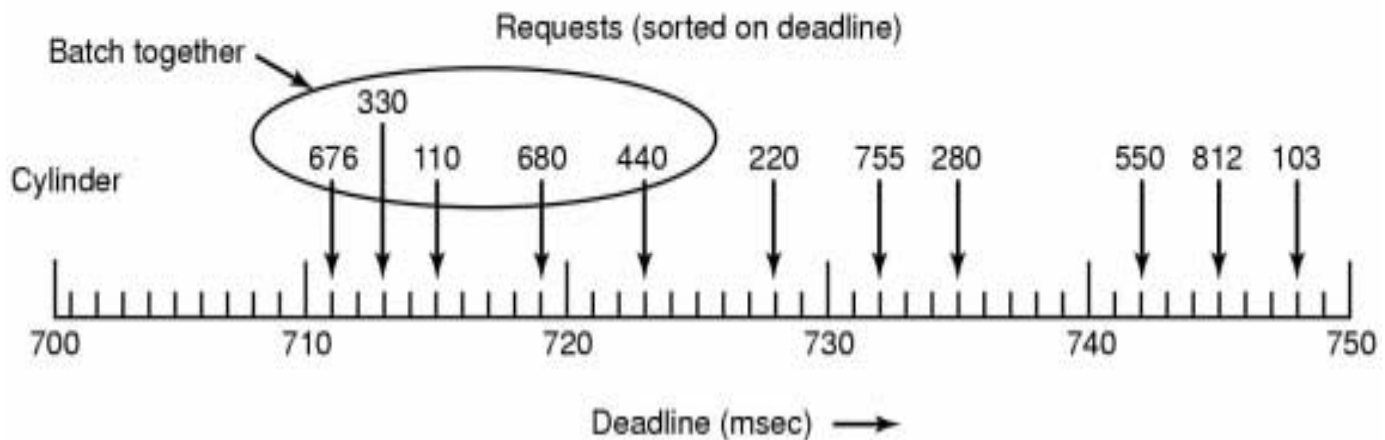
Disk Scheduling per Applicazioni Multimediali

- *Disk Scheduling* Dinamico
 - ▶ Quando gli *stream* hanno caratteristiche diverse tra loro le richieste arrivano al disco in modo poco predicibile e quindi è necessario adottare algoritmi diversi
 - ▶ Ogni richiesta di lettura deve specificare il blocco voluto e la *deadline*
 - ▶ In questo modo è possibile tenere in considerazione il numero di cilindro (come nell'algoritmo dell'ascensore) e la *deadline* (come in EDF)
 - ▶ Si ha il cosiddetto algoritmo *scan-EDF*
 - Vengono servite per prime le k richieste con deadline più prossima, con un ordine che si basa sul numero di cilindro



Disk Scheduling per Applicazioni Multimediali

- *Disk Scheduling Dinamico*



- ▶ Una scelta critica è quella relativa all'ammissione di nuovi utenti
 - E' necessario basarsi sulla stima delle risorse necessarie al nuovo utente considerando eventualmente anche le caratteristiche del film richiesto



Conclusioni

- Riferimenti

- ▶ *Modern Operating Systems* (2^a edizione), A. Tanenbaum
 - Capitolo 7, *Multimedia Operating Systems*