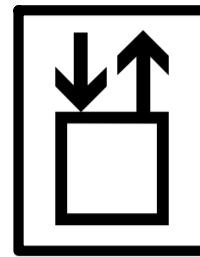


## Algoritmo dell'ascensore: **Scheduling SCAN** (elevator algorithm)

venerdì 21 aprile 2023 12:43



### Scheduling SCAN (elevator algorithm)

L'algoritmo dell'ascensore (anche detto SCAN) è un algoritmo di disk scheduling per stabilire l'ordine in cui devono essere processate le richieste di lettura e richieste di scrittura su disco rigido.

L'algoritmo funziona in maniera simile al shortest seek time first (SSTF) ma, a differenza di quest'ultimo, viene scelto il cilindro con il minore seek time solo nella direzione in cui si sta muovendo la testina, la quale si sposta continuamente avanti e indietro attraversando tutte le tracce del disco.

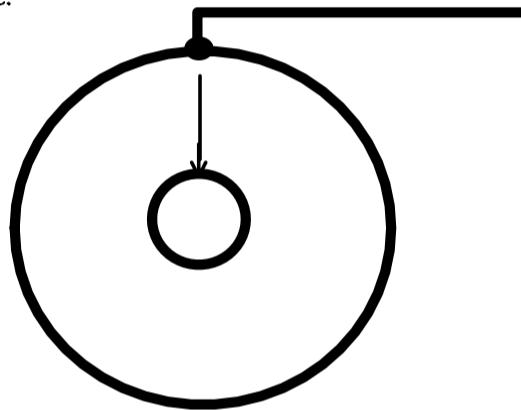
Il nome dell'algoritmo (in inglese **elevator algorithm**) deriva proprio da questo movimento bidirezionale.

Se arriva una nuova richiesta mentre l'unità di I/O è inattiva, il movimento iniziale della testina sarà in direzione della traccia in cui i dati sono/saranno memorizzati. Appena arrivano ulteriori richieste, vengono elaborate solo se compatibili con la direzione corrente del movimento del braccio, fino a quando la testina non raggiunge il bordo (interno o esterno) del disco. Quando ciò accade, la direzione del braccio viene invertita, e le richieste rimaste in attesa vengono servite.

Il seek (il movimento della testina) avviene solo in una data direzione fino al termine delle tracce o fino a che non ci sono più richieste che devono essere servite su tracce richieste in quella direzione.

Supponiamo di avere un hard drive in cui la testina è posizionata sulla traccia esterna, e decidiamo di servire una richiesta che ci permette di spostarci il meno possibile con la testina. Ma quando noi decidiamo è possibile che la sequenza di richieste cambi nel tempo mentre serviamo una richiesta. Se arrivano altre richieste inerenti a tracce che sono già state attraversate da questo ascensore, non c'è verso, quelle tracce ASPETTERANNO.

Tra tutte le richieste che dobbiamo ancora processare andiamo sempre a selezionare le prossime in base al movimento della testina verso il centro dell'hard-drive.



Ovviamente se non ci sono più cose da fare e non ci sono più piani da servire con questo ascensore mentre sale verso il centro, allora cominceremo a riscendere per servire tracce che stavano aspettando e che avevamo attraversato originariamente.

Non tutti possono passare avanti a tutti perché dipende da qual è il movimento attuale dell'ascensore. Se per servire una richiesta che è giunta successivamente dovremmo invertire il movimento dell'ascensore, non lo facciamo ora, anche se quella traccia è molto vicina in base a dove l'ascensore si trova.

Questo algoritmo sfavorisce l'area attraversata più di recente. Se arrivano richieste precedenti rispetto a dove ci troviamo, esse aspetteranno.

- il seek avviene in una data direzione fino al termine delle tracce o fino a che non ci sono più richieste in quella direzione
- sfavorevole all'area attraversata più di recente (ridotto sfruttamento della località)
- la versione C-SCAN utilizza scansione in una sola direzione (fairness nel servizio delle richieste)

### C-SCAN quando scende non porta persone.

L'ascensore è a quota 100, quindi dobbiamo salire e il più vicino è 150.

#### Un esempio

Traccia iniziale = 100

Direzione iniziale = crescente

Insieme delle tracce coinvolte = 55 – 58 – 39 – 18 – 90 – 160 – 150 – 38 – 184

Riordino in base alla direzione di seek = 150 – 160 – 184 – 90 – 58 – 55 – 39 – 38 – 18

distanze 50 – 10 – 24 – 94 – 32 – 3 – 16 – 1 – 20

|insiemedist|

lunghezza media di ricerca

$$\frac{\sum_{i=1}^{|insiemedist|} dist_i}{|insiemedist|} = 27.8$$

Poi infine arriviamo a 184 che è quella a più su di tutte, ora cominciamo a scendere e la prima che incontriamo in discesa è la richiesta 90, poi 58, poi 55, poi 39, 38 e infine 18.

#### Politica:

- in ogni istante, all'ascensore è associata una **direzione corrente** (SU, GIU) e una **posizione corrente** (piano associato alla prossima richiesta da servire).
- ogni richiesta è caratterizzata da una **destinazione T** (che individua il piano di destinazione richiesto dall'utente in attesa):

- se l'ascensore sta salendo (direzione SU), verranno servite TUTTE le richieste con T raggiungibile in quella direzione ( $T > \text{posizione corrente}$ )
- se l'ascensore sta scendendo (direzione GIU), verranno servite tutte le richieste con T raggiungibile in quella direzione ( $T < \text{posizione corrente}$ ).
- le richieste sono servite secondo l'ordine di vicinanza alla richiesta corrente.
- Quando nella direzione scelta non ci sono più richieste da servire, la direzione dell'ascensore viene invertita ed il procedimento è ripetuto.

**N.B.** Viene servita una richiesta alla volta.

Il problema è che noi possiamo fermare l'ascensore su un piano: supponiamo di aver spostato la testina sulla traccia X-esima del nostro Hard Drive, è possibile che nell'insieme delle richieste che arrivano successivamente ci siano altre richieste di I/O che devono lavorare sulla stessa traccia X? SI!

E se abbiamo una sequenza di richieste, non solo formate da X, beh tutte quelle marcate sulla traccia X passano davanti a tutte le altre in questo algoritmo. Perché stanno allo stesso piano di dove l'ascensore già si trova. Quindi l'ascensore rimane fermo lì per servire tutte queste richieste.

Potremmo rimanere fermi in questo piano X-esimo per un tempo non meglio identificato.

Per ovviare a questo problema esiste la schedulazione FSCAN.