

Systèmes d'Exploitation

Cours 9/12 : Stockage externe

Thomas Lavergne

Université Paris-Sud

Licence 3 - semestre S5

Plan

- 1 Structure de disque
 - Rappels
 - Tampon
 - Codes correcteurs
 - Formatage
- 2 Disques durs
- 3 Ordonnancement
- 4 Stockage sur bande
- 5 Conclusion

Rappels

Système de fichiers

- Découpage des fichiers en **blocs logiques**
- **Allocation** des blocs sur le support **physique**

Géré au niveau de contrôleur de périphérique

Problèmes

- Accès aux blocs
 - *Minimiser le temps de réponse du périphérique*
 - *Dépend du matériel et des algorithmes*
- Garantir l'intégrité des données
 - *Vérifier les secteurs*

Tampon

Principe

Le disque est beaucoup plus lent que la RAM

- Ne pas bloquer le processeur pendant le chargement des blocs
→ **tampon** et **transfert par bloc**

Tampon

Principe

Le disque est beaucoup plus lent que la RAM

- Ne pas bloquer le processeur pendant le chargement des blocs
→ tampon et transfert par bloc

Tampon

Espace de stockage plus petit mais plus rapide conservant les données les plus utilisées

Tampon

Principe

Le disque est beaucoup plus lent que la RAM

- Ne pas bloquer le processeur pendant le chargement des blocs
→ tampon et transfert par bloc

Tampon

Espace de stockage plus petit mais plus rapide conservant les données les plus utilisées

Utilisation

- Gestion : cf. algos remplacement de pages
- Utilisé pour pages et blocs disques

Intégrité des données

Détection des secteurs défectueux

Secteur défectueux \rightarrow relecture \neq écriture

Intégrité des données

Détection des secteurs défectueux

Secteur défectueux \rightarrow relecture \neq écriture

Code d'erreur

Fonction de l'ensemble des données du bloc

- Stocké sur le secteur
- Comparé avec $\phi(\text{données_secteur})$

Intégrité des données

Détection des secteurs défectueux

Secteur défectueux \rightarrow relecture \neq écriture

Code d'erreur

Fonction de l'ensemble des données du bloc

- Stocké sur le secteur
- Comparé avec $\phi(\text{données_secteur})$

Exemple : Somme de contrôle

2 bits de données \rightarrow 1 bit de **parité de la somme**

- 0001 1011 0110 1100 $\rightarrow \phi$ 0110 1100
- Vérification : 00**0** \rightarrow ok, 01**0** \rightarrow **erreur**

Intégrité des données

Détection des secteurs défectueux

Secteur défectueux \rightarrow relecture \neq écriture

Code d'erreur

Fonction de l'ensemble des données du bloc

- Stocké sur le secteur
- Comparé avec $\phi(\text{données_secteur})$

Exemple : Somme de contrôle

2 bits de données \rightarrow 1 bit de **parité de la somme**

- 0001 1011 0110 1100 $\rightarrow \phi$ 0110 1100
- Vérification : 000 \rightarrow ok, 010 \rightarrow **erreur**

mais on ne peut pas savoir lequel des 3 bits a été modifié. . .

Codes correcteurs

Principe

- Code d'erreur → détecter secteur défectueux
- Code correcteur → détecter et réparer !

Codes correcteurs

Principe

- Code d'erreur → détecter secteur défectueux
- Code correcteur → détecter et réparer !

Exemple

Tripler toute l'information :

$0 \rightarrow 000$ $1 \rightarrow 111$

Codes correcteurs

Principe

- Code d'erreur → détecter secteur défectueux
- Code correcteur → détecter et réparer !

Exemple

Tripler toute l'information :

$0 \rightarrow 000$ $1 \rightarrow 111$

- 3 bits différents → erreur
- Vote majoritaire → corriger

Codes correcteurs

Principe

- Code d'erreur → détecter secteur défectueux
- Code correcteur → détecter et réparer !

Exemple

Tripler toute l'information :

$0 \rightarrow 000$ $1 \rightarrow 111$

- 3 bits différents → erreur
- Vote majoritaire → corriger

Très coûteux en espace ($\times 3$)!

En résumé

Un secteur contient

- Une **en-tête** utilisée par le contrôleur (ex : numéro de secteur détecté par la tête)
- Les **données** du bloc logique
- Une **terminaison** contenant le code correcteur

En résumé

Un secteur contient

- Une **en-tête** utilisée par le contrôleur (ex : numéro de secteur détecté par la tête)
- Les **données** du bloc logique
- Une **terminaison** contenant le code correcteur

Formatage

Opération de définition des secteurs et des blocs logiques

- Formatage bas niveau : taille des secteurs (et donc des blocs)
- Formatage haut niveau : partition, table FAT

Plan

- 1 Structure de disque
- 2 Disques durs
 - Vitesse angulaire
 - Disque dur
 - Blocs et disques
- 3 Ordonnancement
- 4 Stockage sur bande
- 5 Conclusion

Structures de stockage

Fiches perforées (années 50)

Structures de stockage

Fiches perforées (années 50)

✗ Capacité (nombre de trous)

Bandes magnétiques (années 60)

✓ Capacité

Structures de stockage

Fiches perforées (années 50)

✗ Capacité (nombre de trous)

Bandes magnétiques (années 60)

✓ Capacité

✗ Allocation contiguë

Disques/disquettes (années 70)

✓ Capacité

✓ Allocation libre

✗ Fragilité

Principe

Disque

- Plaque circulaire



Principe

Disque

- Plaque circulaire $\rightarrow n$ pistes concentriques



Principe

Disque

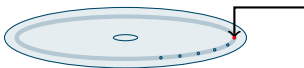
- Plaque circulaire $\rightarrow n$ pistes concentriques
- m secteurs (cadre de blocs) par piste



Principe

Disque

- Plaque circulaire → n pistes concentriques
- m secteurs (cadre de blocs) par piste
- Tête de lecture mobile



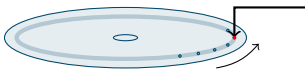
Fonctionnement

- La tête lit 1 secteur à la fois

Principe

Disque

- Plaque circulaire $\rightarrow n$ pistes concentriques
- m secteurs (cadre de blocs) par piste
- Tête de lecture mobile



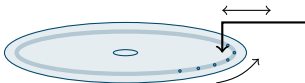
Fonctionnement

- La tête lit 1 secteur à la fois
- Rotation disque \rightarrow lecture des secteurs de la piste

Principe

Disque

- Plaque circulaire → n pistes concentriques
- m secteurs (cadre de blocs) par piste
- Tête de lecture mobile



Fonctionnement

- La tête lit 1 secteur à la fois
- Rotation disque → lecture des secteurs de la piste
- Déplacement tête → lecture des autres pistes

Problème I

Nombre de secteurs par piste

Les pistes concentriques sont toutes de taille différente !

→ Peut-on y mettre un nombre constant de secteurs ?

Problème I

Nombre de secteurs par piste

Les pistes concentriques sont toutes de taille différente !

→ Peut-on y mettre un nombre constant de secteurs ?

Nombre de secteurs variable ?

✗ Difficile à gérer pour le contrôleur !

Ex : conversion bloc logique → adresse physique (piste, secteur)

✗ Temps d'accès variable selon piste !

(plus long pour secteurs extérieurs)

Problème I

Nombre de secteurs par piste

Les pistes concentriques sont toutes de taille différente !

→ Peut-on y mettre un nombre constant de secteurs ?

Nombre de secteurs variable ?

✗ Difficile à gérer pour le contrôleur !

Ex : conversion bloc logique → adresse physique (piste, secteur)

✗ Temps d'accès variable selon piste !

(plus long pour secteurs extérieurs)

→ Le nombre de secteurs est le même pour toutes les pistes
(ils sont plus ou moins espacés)

Problème I

Vitesse de lecture/écriture

- La **vitesse linéaire** détermine le nombre de secteurs par unité de temps.
- La **vitesse angulaire** est la rotation du disque

Vitesse angulaire constante

- Espacement entre secteurs variable
- ✗ Plus difficile pour le **contrôleur**
- ✗ Diminue la vitesse d'accès
- ✓ Rotation constante

Exemple : Disque dur

Problème I

Vitesse de lecture/écriture

- La **vitesse linéaire** détermine le nombre de secteurs par unité de temps.
- La **vitesse angulaire** est la rotation du disque

Vitesse angulaire variable

- Change à chaque piste
- ✗ Plus difficile pour le **matériel**

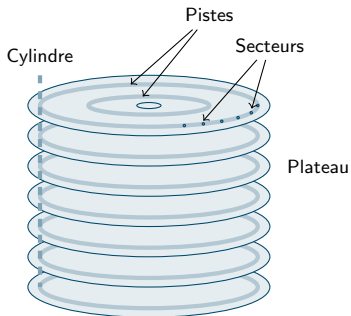
Exemple : CD (car on change moins souvent de piste)

Structure d'un disque dur

Cylindres

Plusieurs disques empilés, appelés **plateaux**

Les pistes de même rayon forment un **cylindre**



Structure d'un disque dur

Disque dur

- Un disque dur est composé de n cylindres
- Chaque cylindre est composé de m pistes
- Chaque piste est composée de k secteurs

Structure d'un disque dur

Disque dur

- Un disque dur est composé de n cylindres
- Chaque cylindre est composé de m pistes
- Chaque piste est composée de k secteurs

Tête de lecture

La tête de lecture est composée de :

- Un bras mobile en **rateau**
- m têtes **fixes** aux extrémités
- Un **multiplexeur** permettant de sélectionner la piste à lire

→ Chaque tête s'insère au dessus d'une piste

Secteurs et blocs

Secteurs

Un **secteur** peut contenir un seul **bloc** de données

Numérotation

- Le bloc 0 est sur le premier secteur de la première piste du cylindre supérieur
- Par secteur croissant, puis par piste croissante, puis par cylindre

Secteurs et blocs

Secteurs

Un **secteur** peut contenir un seul **bloc** de données

Numérotation

- Le bloc 0 est sur le premier secteur de la première piste du cylindre supérieur
- Par secteur croissant, puis par piste croissante, puis par cylindre

Bloc logique

- Adresse physique = (cylindre, piste, secteur)
- Bloc logique → adresse physique

Gestion des secteurs défectueux

Secteur défectueux

Dans tout support physique, certains secteurs deviennent inutilisables avec le temps

Gestion des secteurs défectueux

Secteur défectueux

Dans tout support physique, certains secteurs deviennent inutilisables avec le temps

Table des blocs

- Marquer le secteur inutilisable
- Modifier l'adresse physique associée au bloc logique

→ Il faut une table des blocs (association bloc ↔ secteur)

Méthodes de gestion

- Années 90 : disque IDE → par l'OS + marquer secteurs défectueux dans la FAT
- Disques SCSI : contrôleur du périphérique

Gestion des secteurs défectueux II

Structures de données

- Table des secteurs défectueux
- Réserver un **ensemble de secteurs pour des remplacement** lors du formatage du disque

Contrôleur de périphérique

- Vérification à l'écriture
- Préviend l'OS d'un secteur défectueux
- L'OS demande un remplacement (**glissement**)
- Accès transparent pour l'OS qui ne voit que des blocs **logiques**

Plan

- 1 Structure de disque
- 2 Disques durs
- 3 Ordonnancement
 - Problème
 - First Come, First Served
 - Shortest Seek Time First
 - Scan et Look
 - C-scan et C-Look
 - Performance
- 4 Stockage sur bande

Problème

Accès à un secteur

- Positionnement de la tête de lecture sur la piste
donc sur le bon cylindre...
- Rotation du disque (maximum un tour d'attente)

Problème

Accès à un secteur

- Positionnement de la tête de lecture sur la piste
donc sur le bon cylindre...
- Rotation du disque (maximum un tour d'attente)

Temps d'accès

- Le disque tourne en permanence
 - Vitesse de rotation = caractéristique matérielle
- On peut agir sur les déplacements d'un cylindre à l'autre

Problème

Accès à un secteur

- Positionnement de la tête de lecture sur la piste
donc sur le bon cylindre...
- Rotation du disque (maximum un tour d'attente)

Temps d'accès

- Le disque tourne en permanence
 - Vitesse de rotation = caractéristique matérielle
- On peut agir sur les déplacements d'un cylindre à l'autre

Principe

Minimiser le temps de déplacement de la tête en parcourant les cylindres dans un ordre intelligent

Exemple

Caractéristique disque

- Vitesse de rotation fixée
- 256 cylindres

Exemple

Caractéristique disque

- Vitesse de rotation fixée
- 256 cylindres

Requêtes

Au temps 0, la tête est sur le cylindre 53.

Le contrôleur reçoit une demande d'accès aux cylindres suivants :

98, 183, 37, 122, 14, 124, 65, 67

Exemple

Caractéristique disque

- Vitesse de rotation fixée
- 256 cylindres

Requêtes

Au temps 0, la tête est sur le cylindre 53.

Le contrôleur reçoit une demande d'accès aux cylindres suivants :

98, 183, 37, 122, 14, 124, 65, 67

Ordonnancement optimal

Pas d'autre demande → trier et partir du plus petit (plus proche)

En pratique : la file est **dynamique** → on reçoit d'autres demandes pendant qu'on traite !

First Come, First Served

Principe

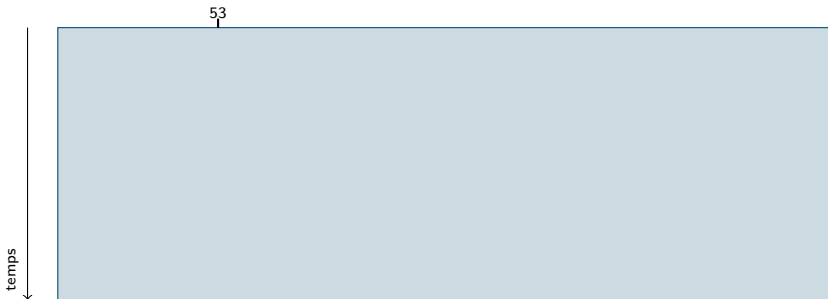
Prendre les cylindres dans l'ordre

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

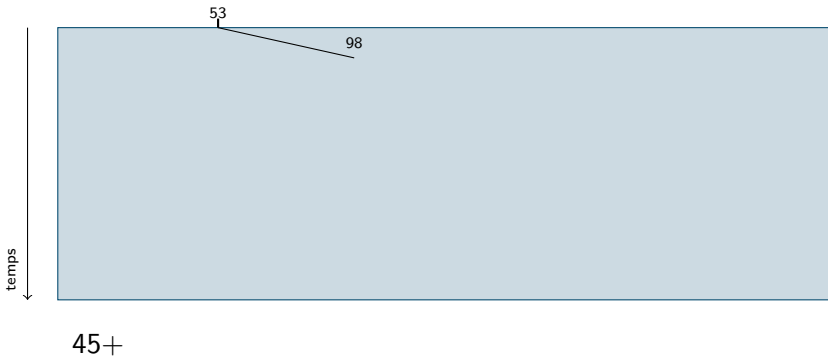


First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

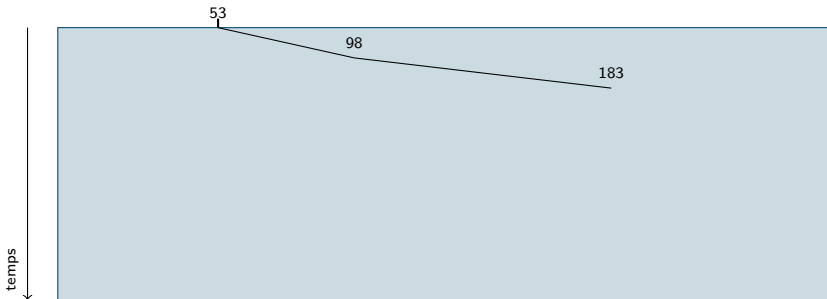


First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



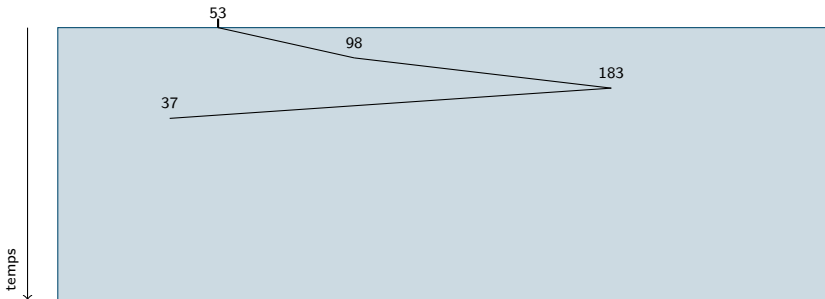
45+85+

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



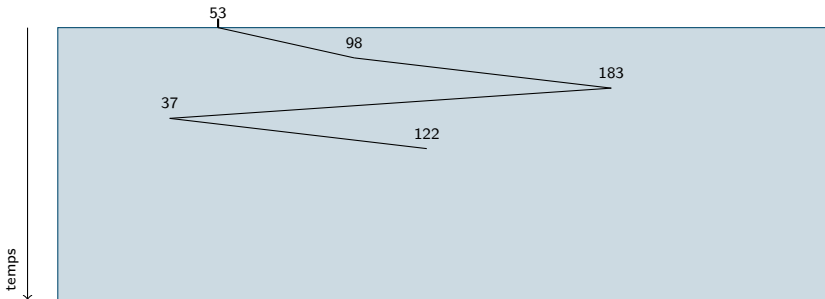
45+85+146+

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



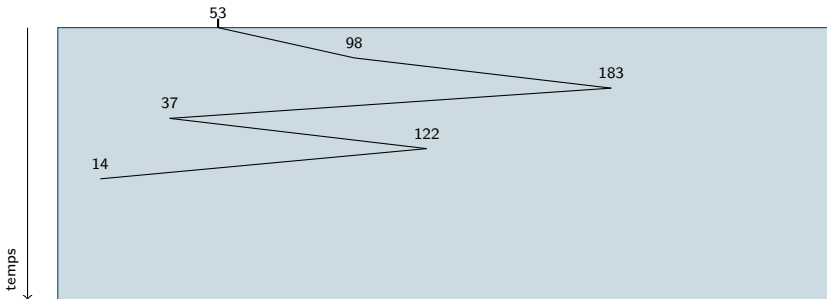
45+85+146+85+

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



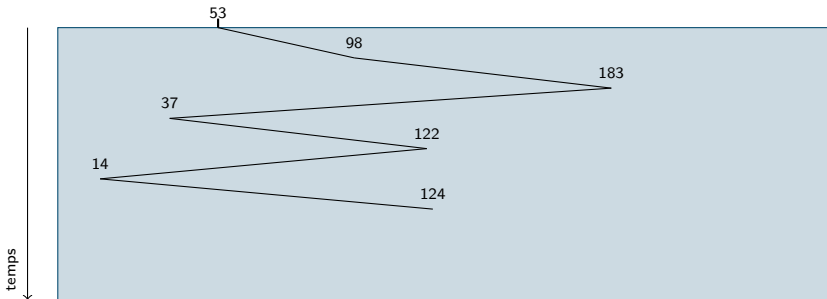
45+85+146+85+108+

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



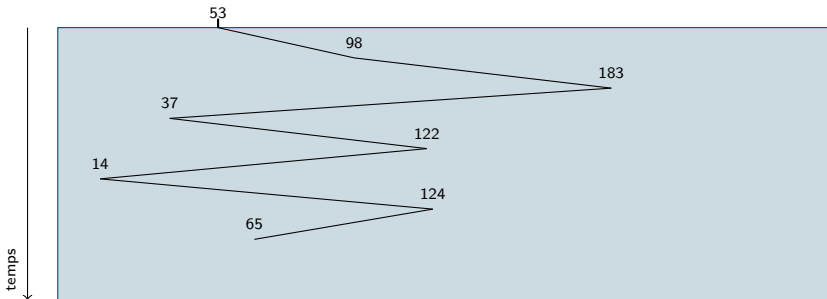
45+85+146+85+108+110+

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



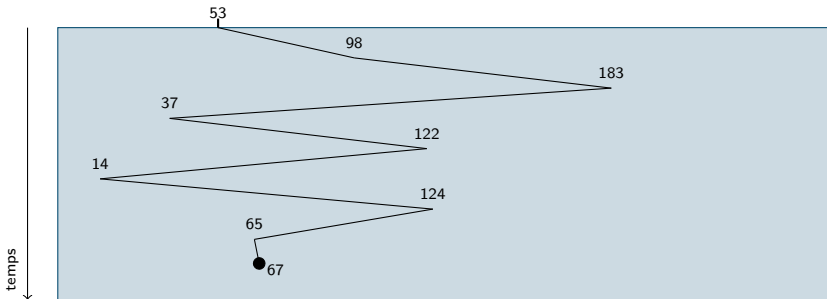
$$45 + 85 + 146 + 85 + 108 + 110 + 59 +$$

First Come, First Served

Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$45 + 85 + 146 + 85 + 108 + 110 + 59 + 2 = 640$ cylindres parcourus

Shortest Seek Time First

Principe

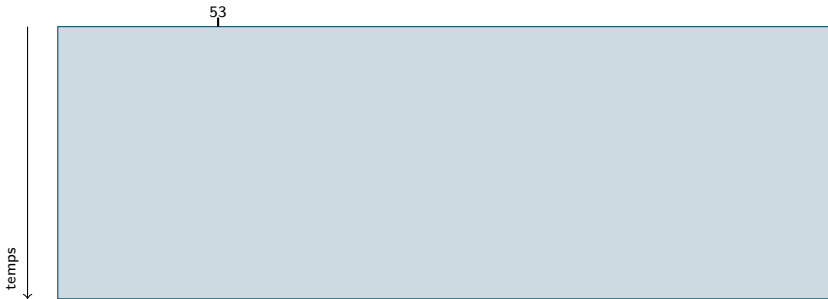
Aller vers le cylindre le plus proche

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

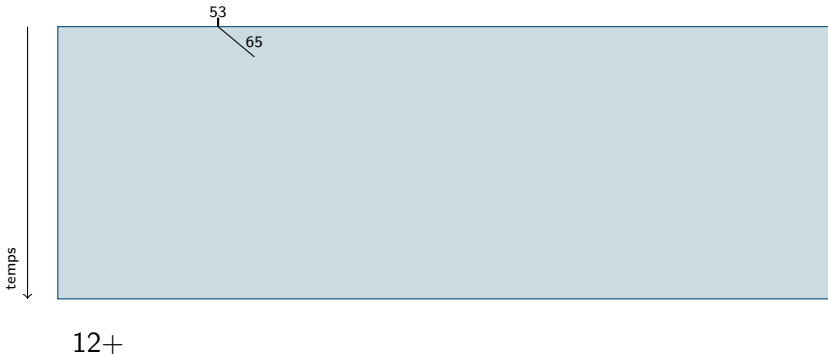


Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

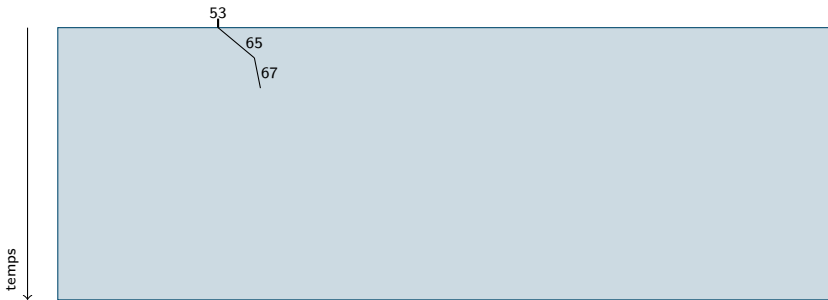


Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



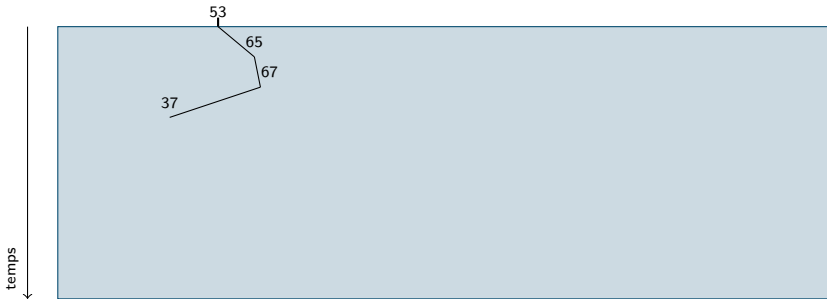
12+2+

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



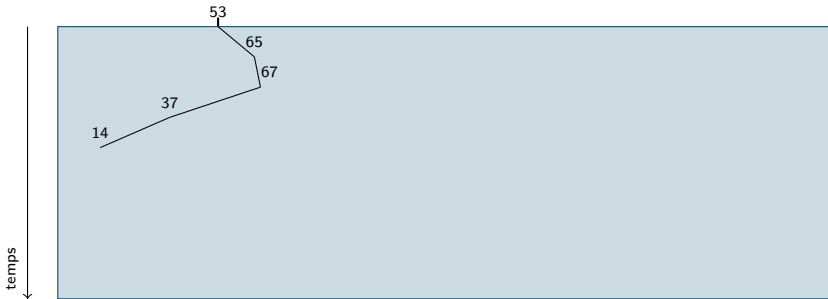
12+2+30+

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



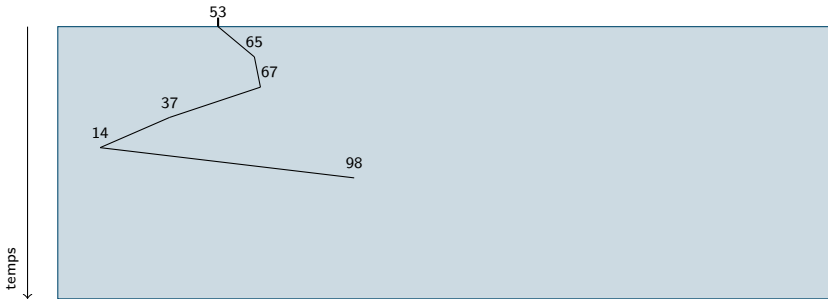
12+2+30+23+

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



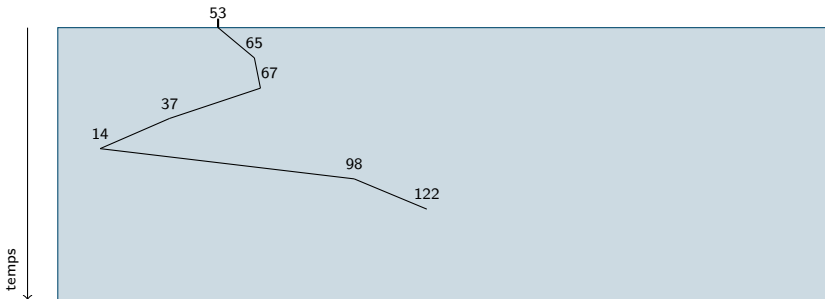
12+2+30+23+84+

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



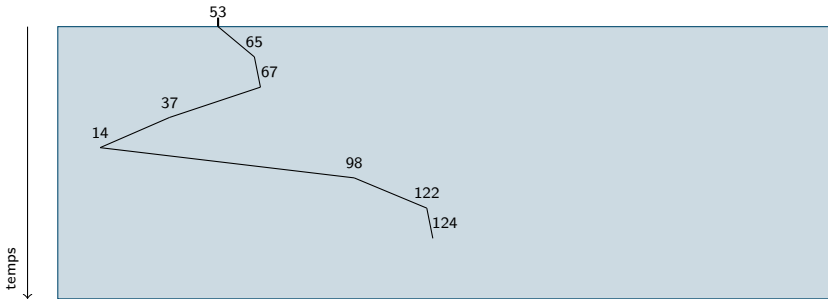
12+2+30+23+84+24+

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



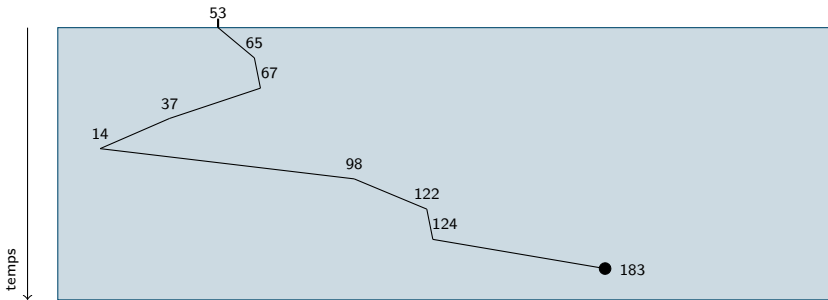
12+2+30+23+84+24+2+

Shortest Seek Time First

Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$12 + 2 + 30 + 23 + 84 + 24 + 2 + 59 = 236$ cylindres parcourus

Shortest Seek Time First

Avantages

Shortest Seek Time First

Avantages

- ✓ Temps de traitement souvent très bon

Limites

Shortest Seek Time First

Avantages

- ✓ Temps de traitement souvent très bon

Limites

- ✗ Pas forcément optimal...

sur l'exemple, en servant 37 en premier, on aurait un meilleur temps

→ Il faut tenir compte de nouvelles arrivées possibles...

Shortest Seek Time First

Avantages

- ✓ Temps de traitement souvent très bon

Limites

- ✗ Pas forcément optimal...

sur l'exemple, en servant 37 en premier, on aurait un meilleur temps

→ Il faut tenir compte de nouvelles arrivées possibles...

- ✗ Risque de **famine** !

Tant qu'il arrive des cylindres proches, on reste dans la zone et les autres cylindres ne sont pas servis !

Scan

Principe

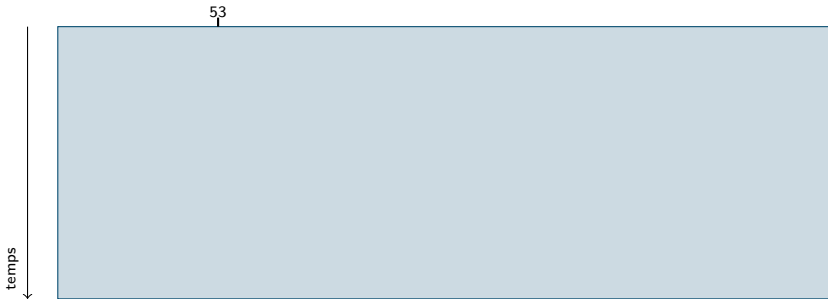
Balayer dans un sens puis dans l'autre

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant

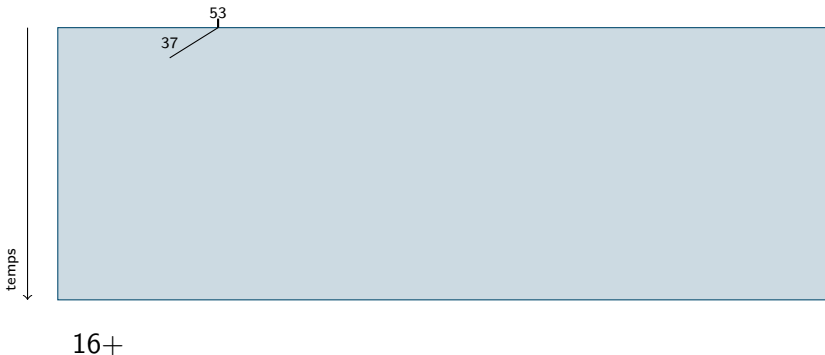


Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant

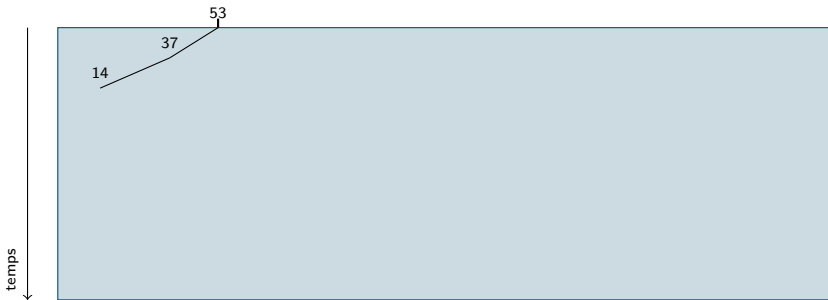


Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



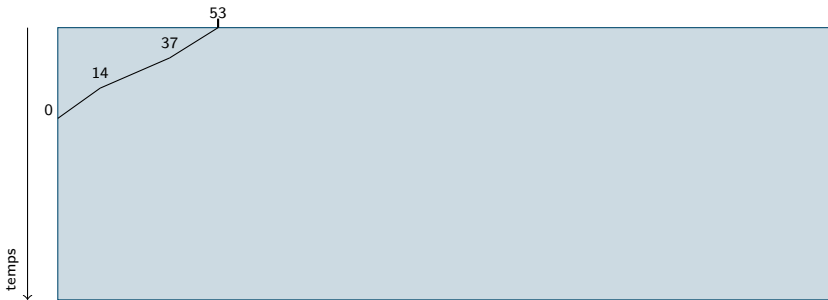
16+23+

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



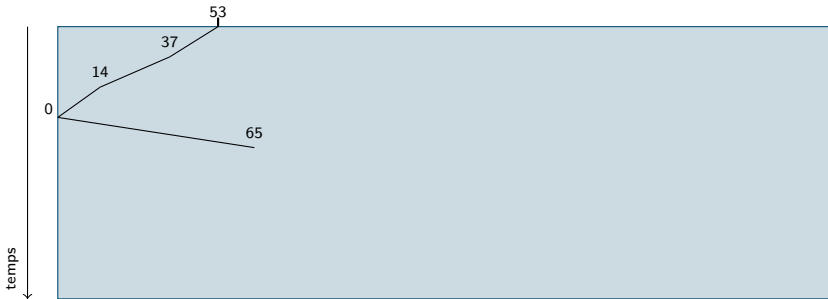
16+23+14+

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



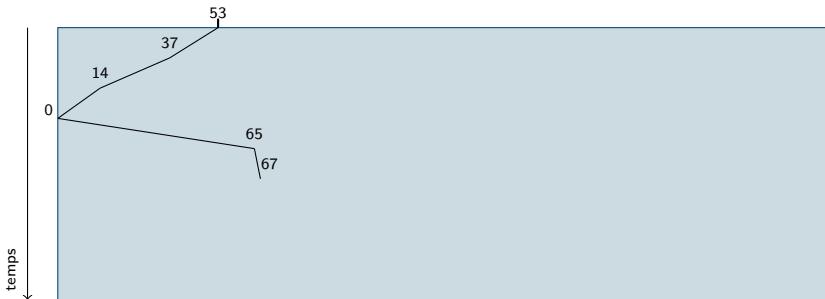
$$16 + 23 + 14 + 65 +$$

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



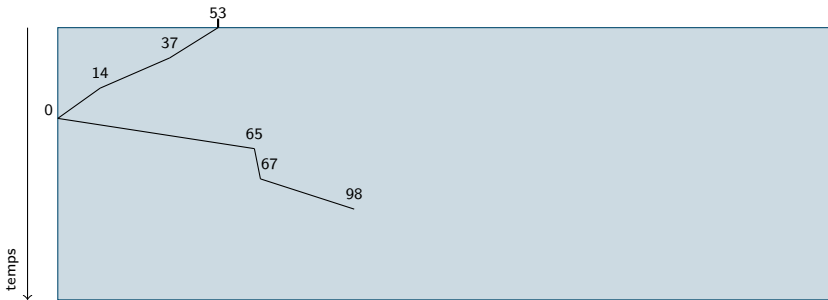
$$16+23+14+65+2+$$

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



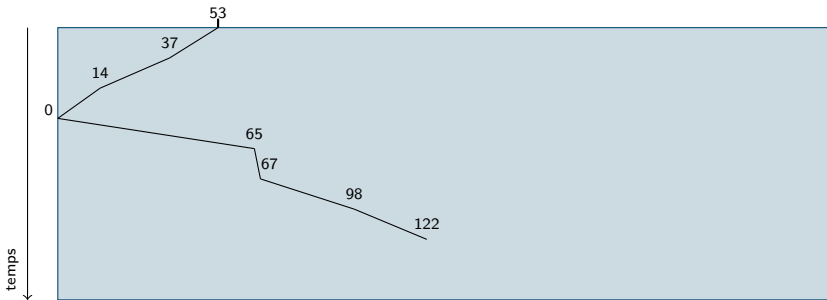
$$16+23+14+65+2+31+$$

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



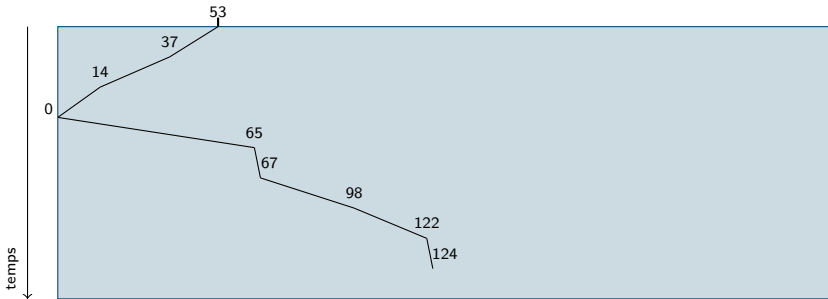
$$16+23+14+65+2+31+24+$$

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



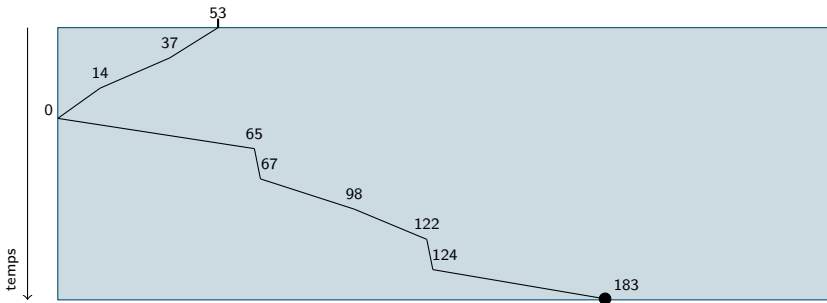
$$16+23+14+65+2+31+24+2+$$

Scan

Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



$$16 + 23 + 14 + 65 + 2 + 31 + 24 + 2 + 59 = 53 + 183 = 236$$

Scan

Avantages

Scan

Avantages

- ✓ Temps de traitement souvent très bon
- ✓ Pas de famine

Limites

Scan

Avantages

- ✓ Temps de traitement souvent très bon
- ✓ Pas de famine

Limites

- ✗ Parcours inutiles vers les bords
- ✗ Lorsqu'on fait demi-tour, on vient de servir les cylindres près du bord → il est peu probable d'en avoir beaucoup à traiter par ici...

même en tenant compte de nouvelles arrivées !

Look

Principe

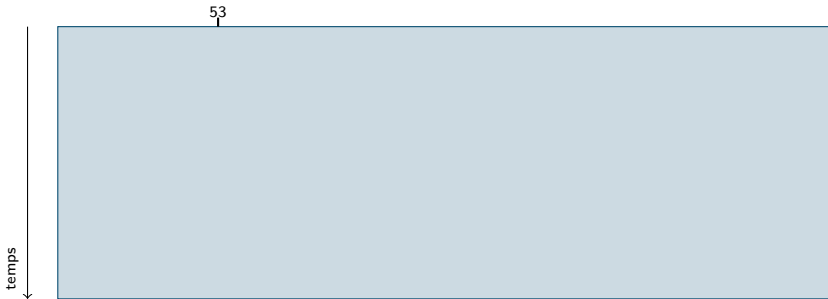
Repartir lorsqu'on a atteint le plus petit cylindre demandé

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant

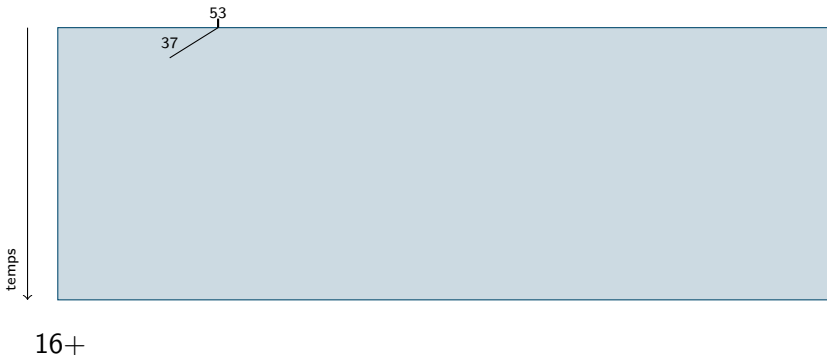


Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant

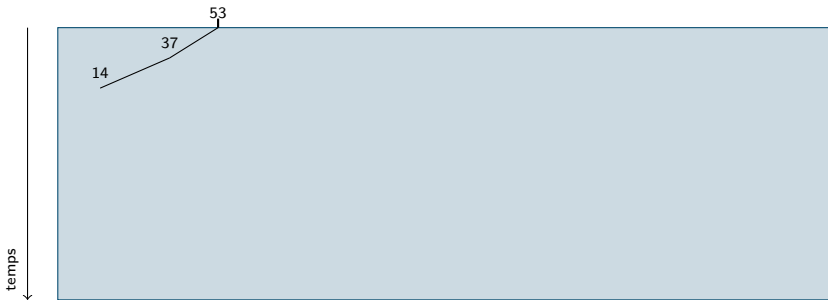


Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



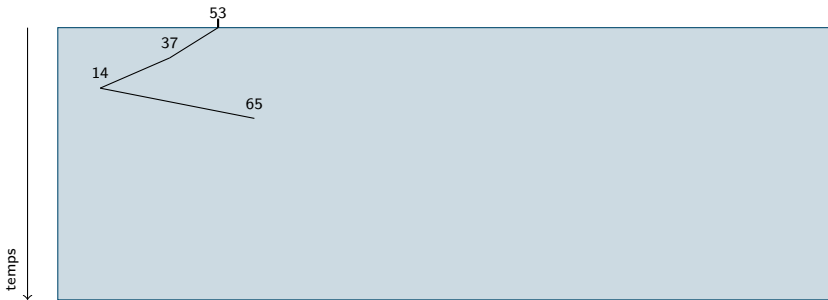
16+23+

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



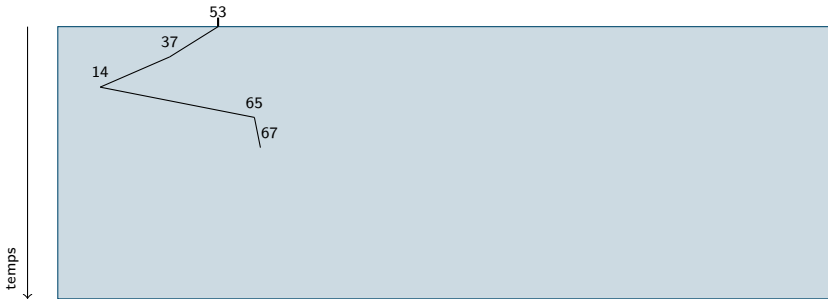
$$16+23+51+$$

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



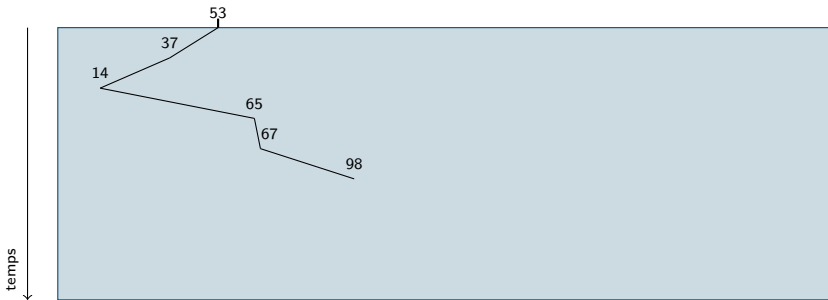
$$16+23+51+2+$$

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



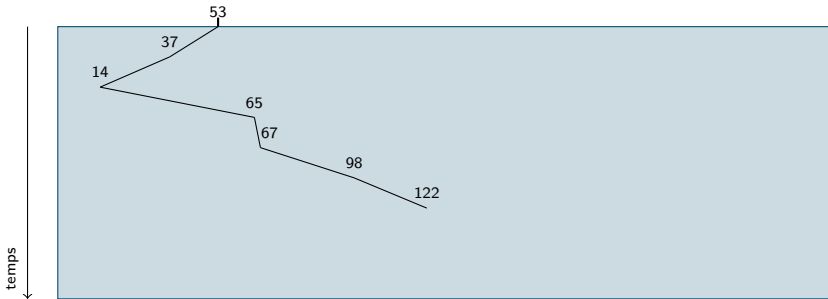
$$16+23+51+2+31+$$

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



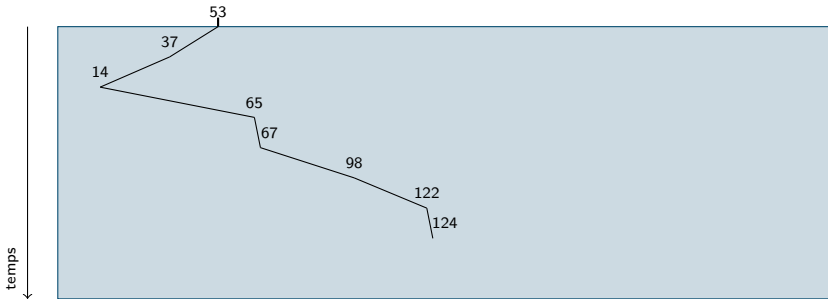
16+23+51+2+31+24+

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



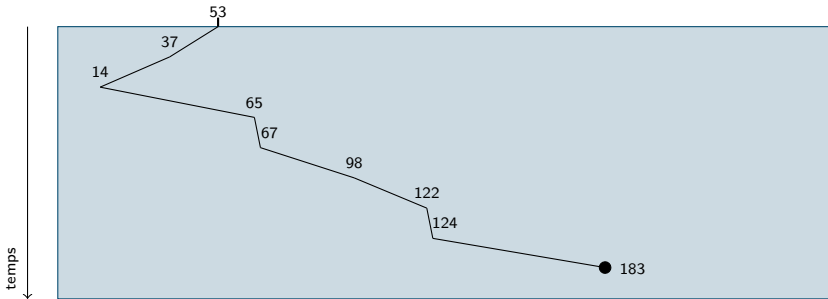
$$16+23+51+2+31+24+2+$$

Look

Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 } sur descendant



$$16+23+51+2+31+24+2+59 = (53-14) + (183-14) = 208$$

C-Scan

Principe

Balayage circulaire : toujours dans le même sens

C-Scan

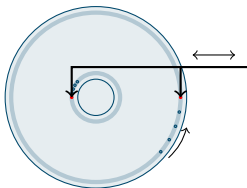
Principe

Balayage circulaire : toujours dans le même sens

Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0, secteur α pendant que tête 2 lit le secteur n , secteur $\alpha + \pi$



C-Scan

Principe

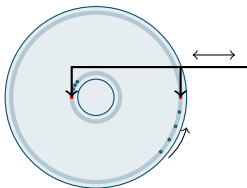
Balayage circulaire : toujours dans le même sens

Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0, secteur α pendant que tête 2 lit le secteur n , secteur $\alpha + \pi$

- ① Extérieur \rightarrow intérieur
 \Rightarrow Tête 1 balaye de 0 à $n-1$



C-Scan

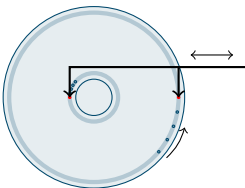
Principe

Balayage circulaire : toujours dans le même sens

Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0, secteur α pendant que tête 2 lit le secteur n , secteur $\alpha + \pi$



- 1 Extérieur \rightarrow intérieur
 \Rightarrow Tête 1 balaye de 0 à $n-1$
- 2 Tête 1 = secteur n
 \Rightarrow Tête 2 = secteur 0

C-Scan

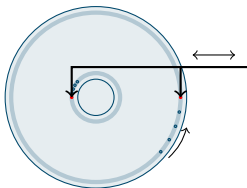
Principe

Balayage circulaire : toujours dans le même sens

Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0, secteur α pendant que tête 2 lit le secteur n , secteur $\alpha + \pi$



- ① Extérieur \rightarrow intérieur
 \Rightarrow Tête 1 balaye de 0 à $n-1$
- ② Tête 1 = secteur n
 \Rightarrow Tête 2 = secteur 0
- ③ Intérieur \rightarrow extérieur
 \Rightarrow Tête 2 balaye de 0 à $n-1$

C-Scan

Principe

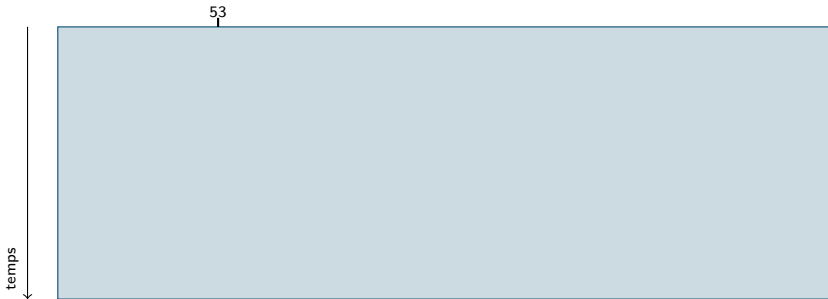
Balayage circulaire (ici, descendant)

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

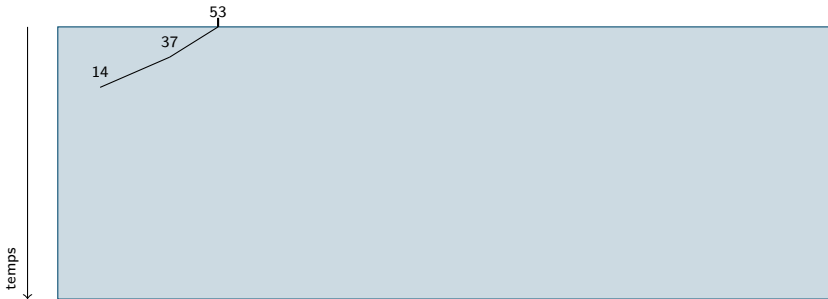


C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



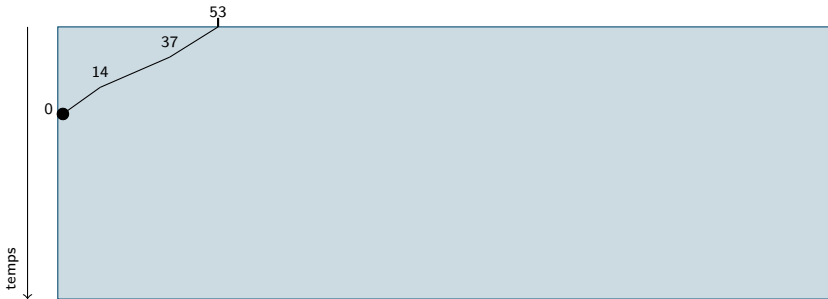
16+23+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



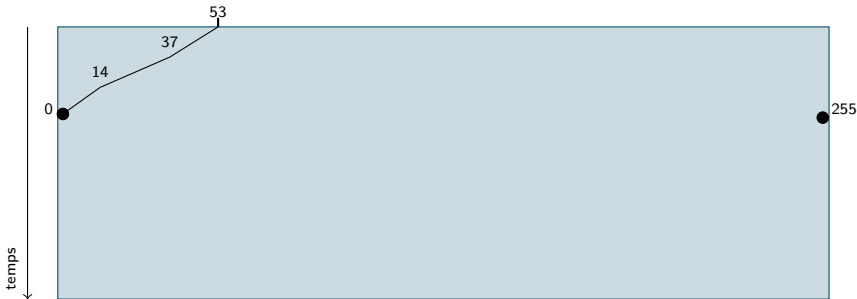
16+23+14+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



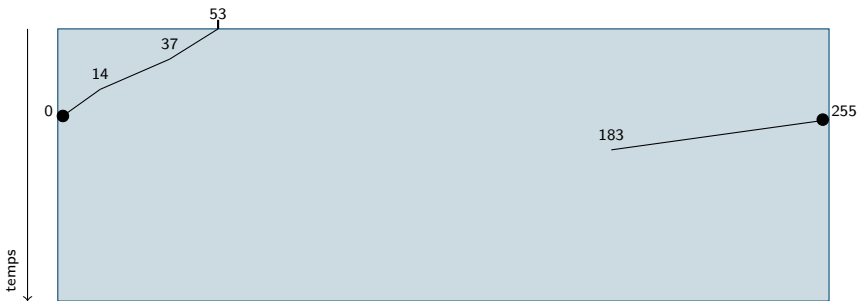
16+23+14+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



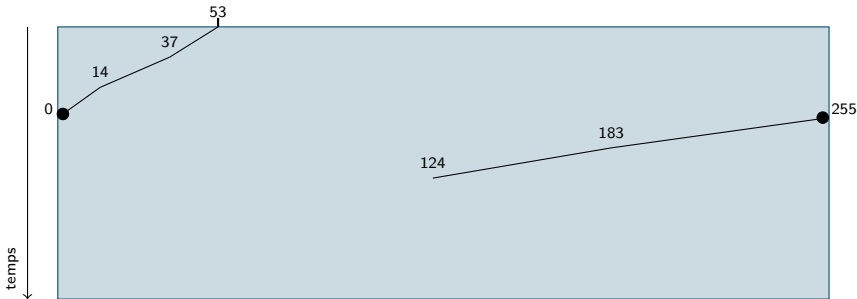
16+23+14+72+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



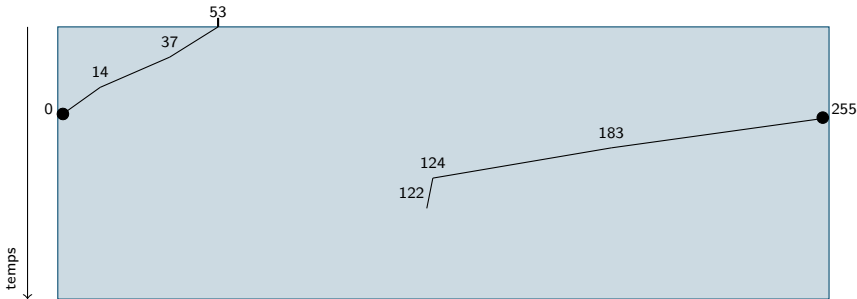
16+23+14+72+59+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



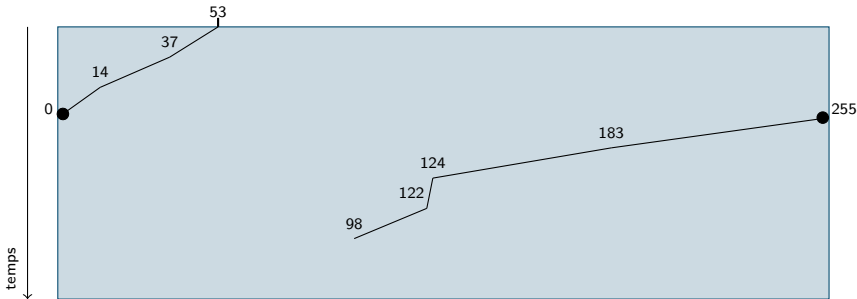
16+23+14+72+59+2+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



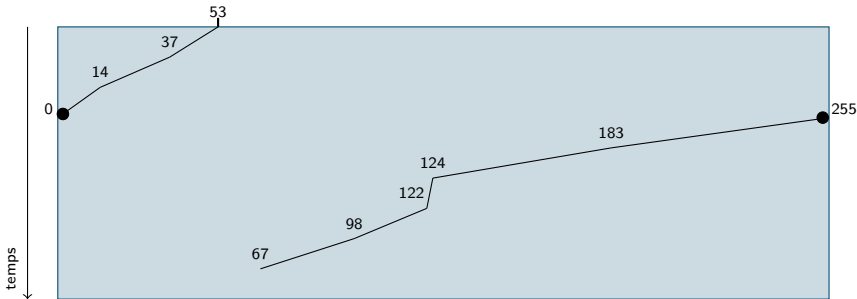
16+23+14+72+59+2+24+

C-Scan

Principe

Balayage circulaire (ici, **descendant**)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



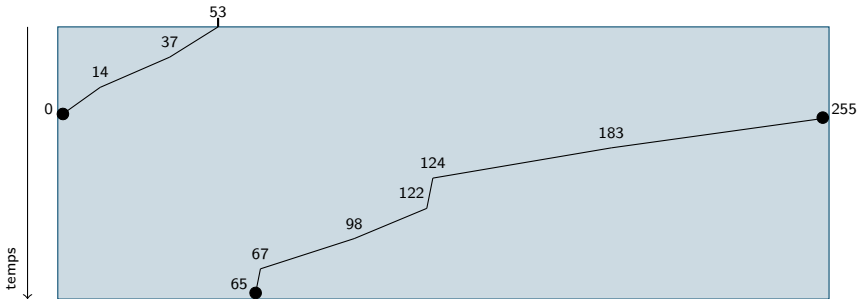
16+23+14+72+59+2+24+31+

C-Scan

Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$$16+23+14+72+59+2+24+31+2 = 53 + (255-65) = 243$$

C-Look

Principe

Même principe sans aller jusqu'au bord :

C-Look

Principe

Même principe sans aller jusqu'au bord :

- 1 Balayer intérieur \rightarrow extérieur, tête 1
 \Rightarrow sens descendant

C-Look

Principe

Même principe sans aller jusqu'au bord :

- 1 Balayer intérieur \rightarrow extérieur, tête 1
 \Rightarrow sens descendant
- 2 Lorsque *min* est atteint, changer de tête
 \Rightarrow sens montant

C-Look

Principe

Même principe sans aller jusqu'au bord :

- 1 Balayer intérieur \rightarrow extérieur, tête 1
 \Rightarrow sens descendant
- 2 Lorsque *min* est atteint, changer de tête
 \Rightarrow sens montant
- 3 Continuer intérieur \rightarrow extérieur jusqu'à *max*
sauf si $max \leq position\ courante$

C-Look

Principe

Même principe sans aller jusqu'au bord :

- ➊ Balayer intérieur \rightarrow extérieur, tête 1
 \Rightarrow sens descendant
- ➋ Lorsque *min* est atteint, changer de tête
 \Rightarrow sens montant
- ➌ Continuer intérieur \rightarrow extérieur jusqu'à *max*
sauf si $max \leq position\ courante$
- ➍ Lorsque *max* est atteint, changer de sens
 \Rightarrow sens descendant à nouveau

C-Look

Principe

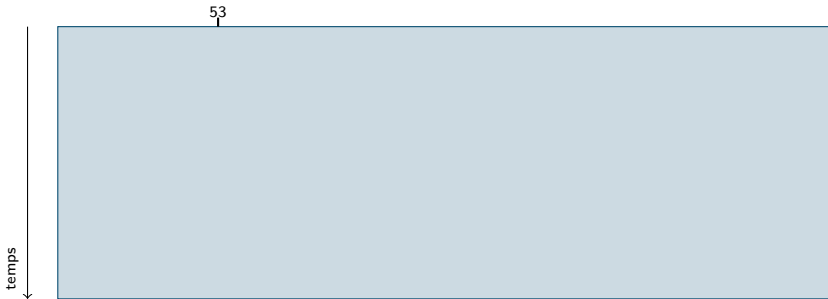
Même principe sans aller jusqu'au bord

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

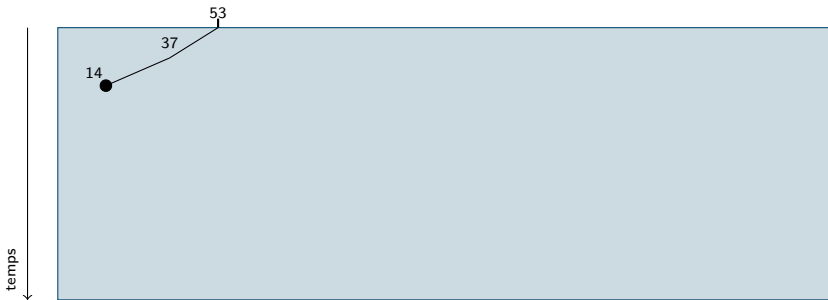


C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



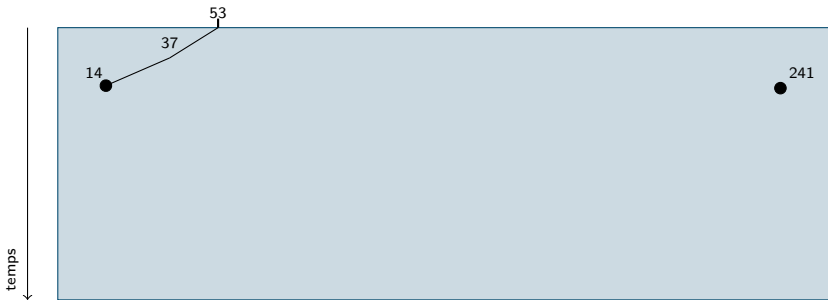
16+23+

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



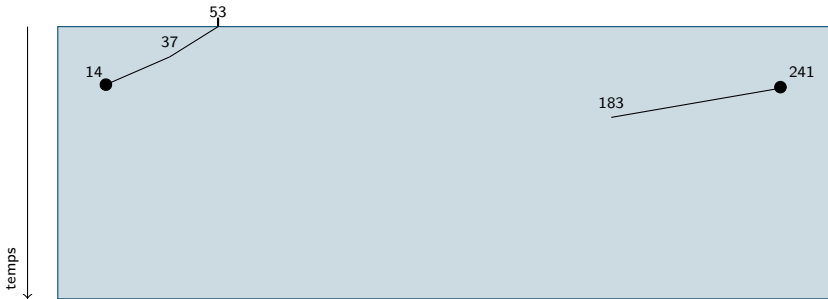
16+23+

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



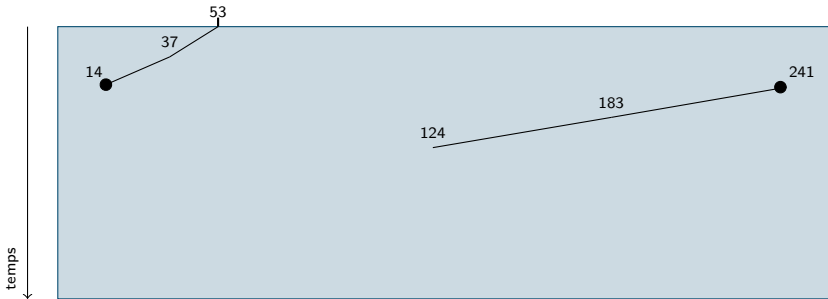
$$16+23+58+$$

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



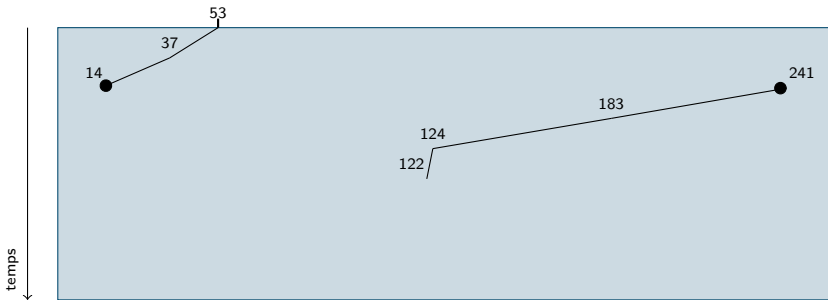
16+23+58+59+

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



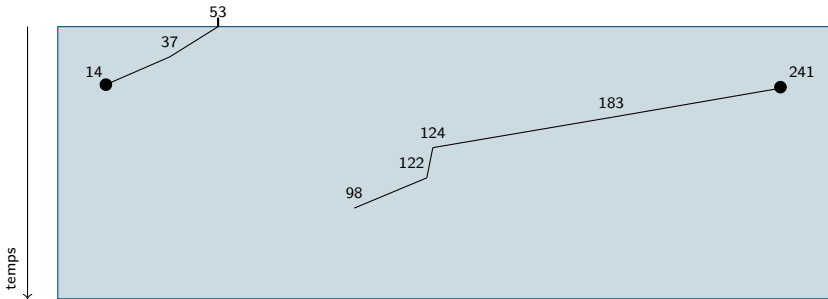
16+23+58+59+2+

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



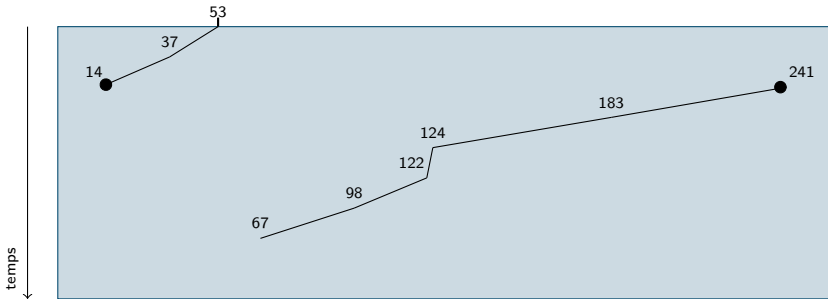
16+23+58+59+2+24+

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



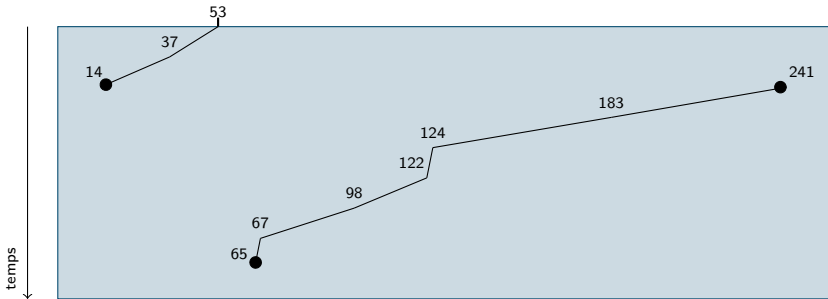
16+23+58+59+2+24+31+

C-Look

Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$$16+23+58+59+2+24+31+2 = (53-14) + (241-65) = 215$$

Performance

En pratique

- La plupart des OS et utilisent **SSTF**
- Les OS mobilisant **beaucoup** le disque utilisent **C-LOOK**

Ordonnancement optimal

Possible à calculer à chaque pas de temps mais très coûteux

Performance

En pratique

- La plupart des OS et utilisent **SSTF**
- Les OS mobilisant **beaucoup** le disque utilisent **C-LOOK**

Ordonnancement optimal

Possible à calculer à chaque pas de temps mais très coûteux

Rappel

Le temps de réponse dépend aussi de :

- La méthode d'**allocation de fichiers** (contiguë, indexée...)
- La position des répertoires et des blocs d'index
- Le temps de rotation du disque
- La priorité au niveau OS (pagination vs E/S)

Plan

- 1 Structure de disque
- 2 Disques durs
- 3 Ordonnancement
- 4 Stockage sur bande**
- 5 Conclusion

Stockage sur bande magnétique

Avantages

- ✓ Coût : **très peu cher** au Tio ! (facteur 5 à 10)
 - Stockage préventif de données brutes (**Big Data**)
 - ✓ Ex : 15 To pour 40 EUR en 2020
- ✓ Coût de **maintenance** quasi nul
- ✓ **Durée de vie** (sans perte de données) plus élevée
(CD = 5 ans, DD = 5 à 10 ans, Bande = 20 à 30 ans)
- ✓ Volume (ex : LTO-8 → 150 Go/cm³)

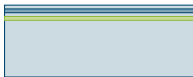
Inconvénients

- ✗ Temps d'accès aléatoire (lecture et écriture) très élevé
- Réservé au stockage à froid ou application spécifiques

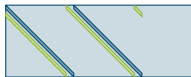
Structures

Pistes

- 9 pistes : 8 données + 1 parité



pistes linéaires



pistes hélicoïdales

Structures

Pistes

- 9 pistes : 8 données + 1 parité

Blocs

- Secteurs taille fixe
- Intervalles inter-enregistrement (IRG) (entre 2 secteurs)
 - ➔ On s'arrête uniquement sur les IRG
 - ➔ Interruption sur secteur → rembobiner à l'IRG précédent

Structures

Pistes

- 9 pistes : 8 données + 1 parité

Blocs

- Secteurs taille fixe
- Intervalles inter-enregistrement (IRG) (entre 2 secteurs)
 - ➔ On s'arrête uniquement sur les IRG
 - ➔ Interruption sur secteur ➔ rembobiner à l'IRG précédent

Avantages

- Lecture possible dans les 2 sens
- Stockage en baies

Plan

- 1 Structure de disque
- 2 Disques durs
- 3 Ordonnancement
- 4 Stockage sur bande
- 5 Conclusion**

Ce qu'il faut retenir

- Stockage externe : pistes, secteurs
Blocs logiques → secteurs
- Disque dur : cylindres, pistes, secteurs
- Gestion des blocs défectueux
- Codes d'erreur et codes correcteurs
- Ordonnancement d'accès aux secteurs
 - FCFS
 - SSTF
 - Scan et Look
 - C-scan et C-look
- Stockage sur bande