



Systeme d'Exploitation 2

Chapitre 3 : La gestion de la mémoire secondaire

(partie 1)

1 ère Année Second Cycle

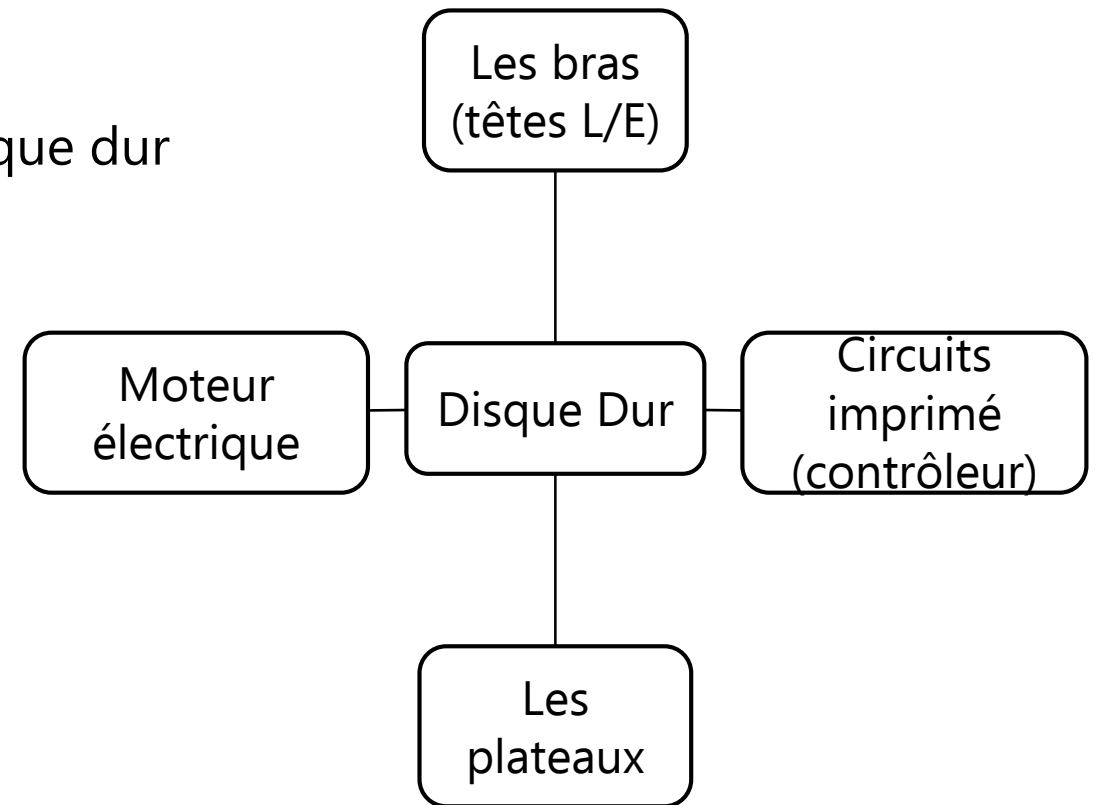
Dr. M. Baba Ahmed

Rappel : la mémoire de masse (Disque dur)

- Composant principal d'une machine
- Stockage de données persistantes (SE, données, logiciels ...) (non volatile)
- Mise en œuvre d'une mémoire virtuelle (swap)
- Capacité : jusqu'à 8 To
- Taux de transfert : vitesse lecture/écriture de 40 Mo/s à 260 Mo/s
- Vitesse de rotation : 7200 à 15000 rpm tours par minute

Rappel : la mémoire de masse (Disque dur)

- Temps d'accès moyen : le temps que met la tête pour se positionner sur la bonne piste (de 3 à 12 ms)
- Mémoire cache : les données auxquelles le disque dur accède souvent (amélioration des performances)
- Interface : connectique du disque dur
IDE (Integrated Drive Electronics),
S-ATA (Serial Advanced Technology Attachment)



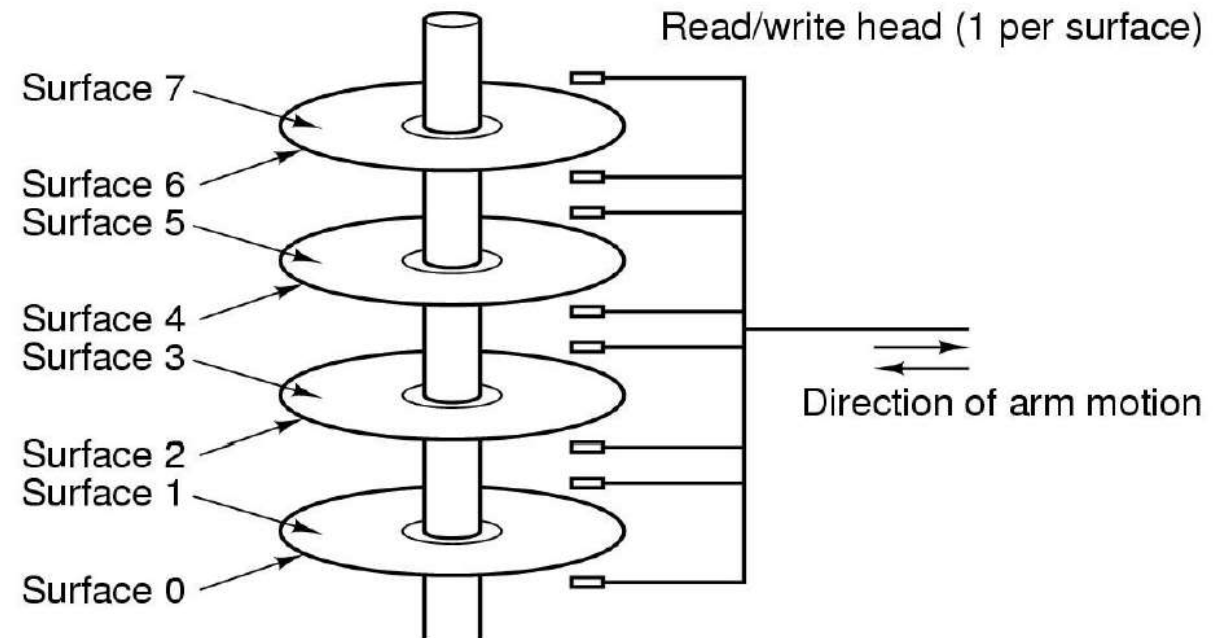
Rappel : la structure d'un Disque dur

Logiquement :

- Suite de blocs contigüe
- Le bloc est la plus petite unité de transfert vers la mémoire centrale

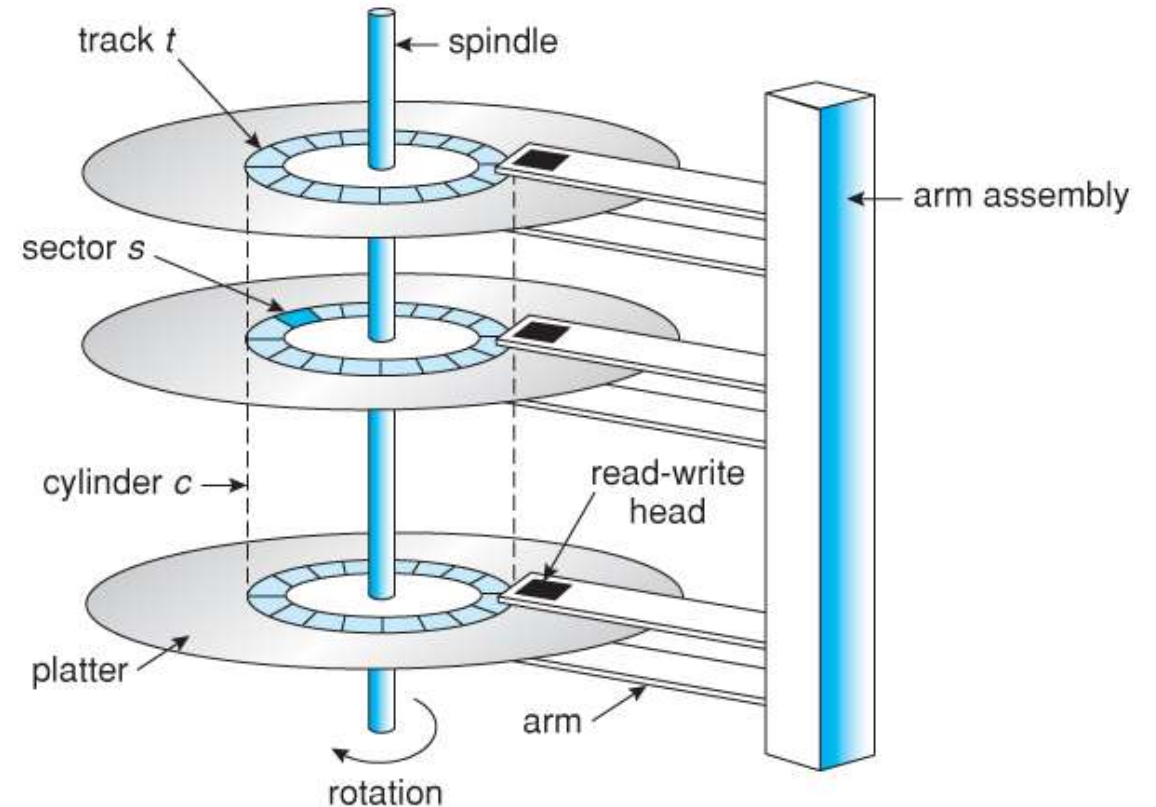
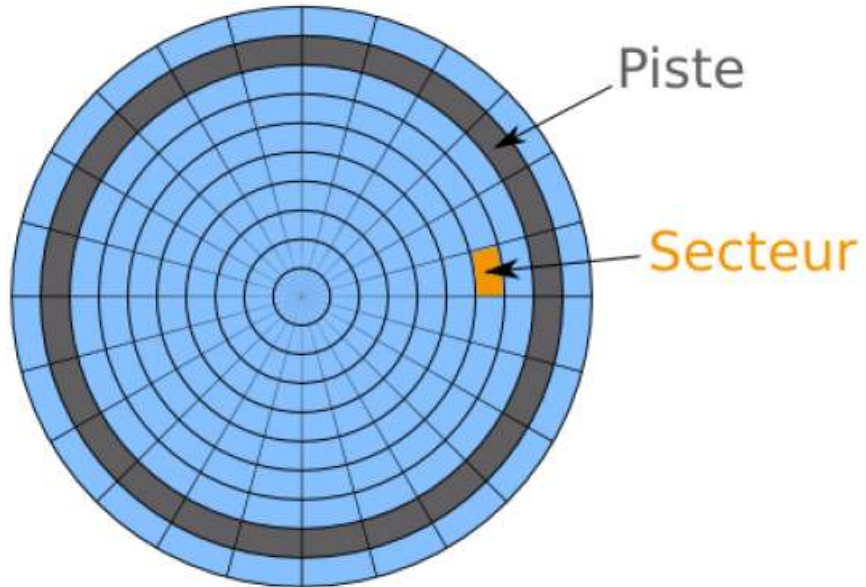
Physiquement :

- Ensemble de plateaux (double faces)
- Chaque plateau est composé de pistes
- Les pistes parallèles forment un cylindre.
- Chaque piste est composée de secteurs



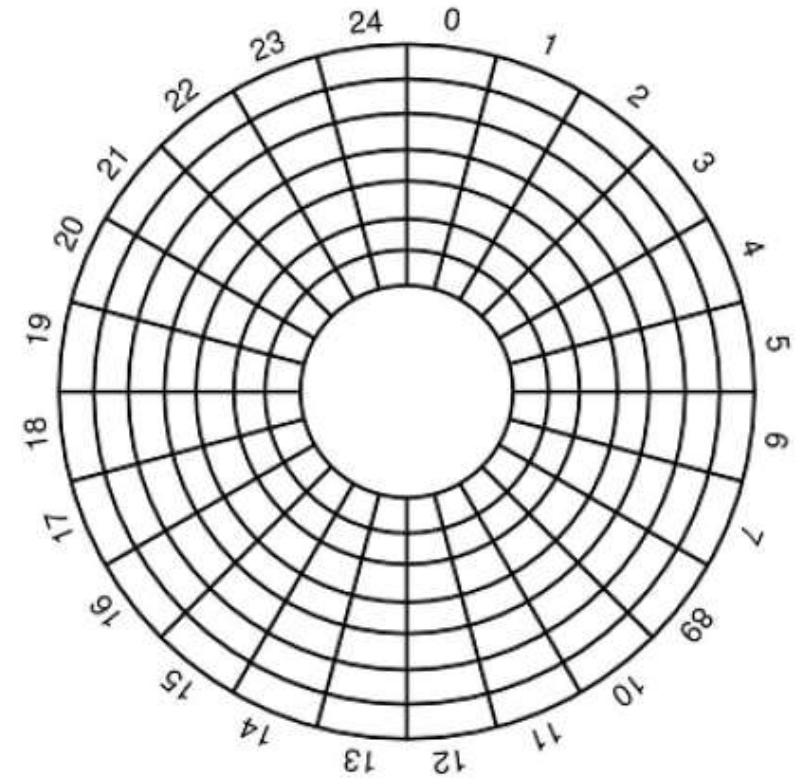
Rappel : la structure d'un Disque dur

Cylindre : l'ensemble de pistes qui se trouvent dans la même position du bras de lecture/écriture



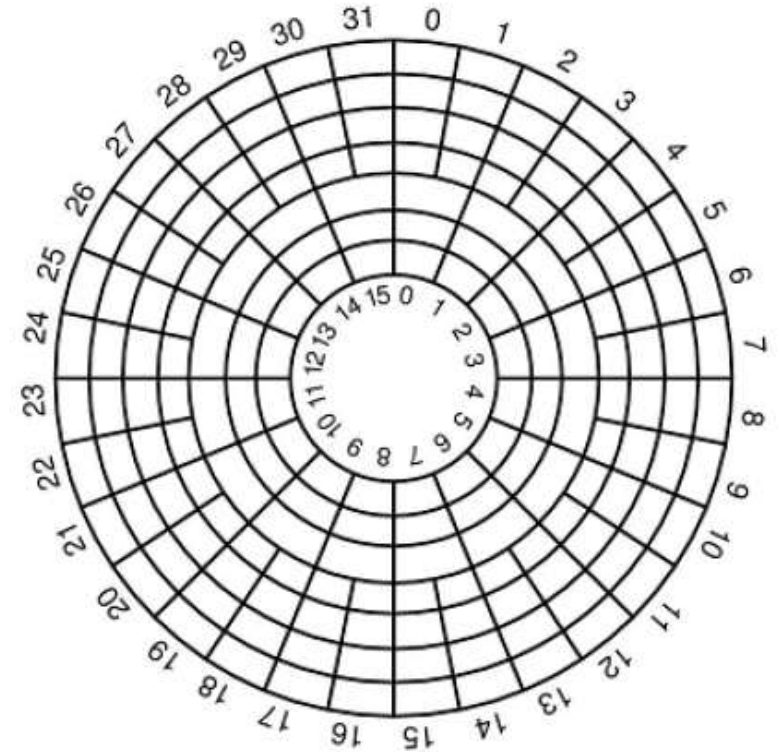
Pistes et secteurs

- Sur les vieux disques, le nombre de secteurs par piste étaient constant pour tout les cylindres
- La densité d'enregistrement était maximale vers le centre mais plus faible vers l'extérieur, là où les pistes sont plus longues.
- Le nombre de cylindres, de têtes et de secteurs forment la géométrie du disque.
- Ces paramètres notés CHS (*Cylinder- Head - Sector* / *Cylindre - tête - secteur*) sont généralement indiqués sur le couvercle du boîtier.



Pistes et secteurs

- La structure CHS des disques durs induit un gaspillage du fait d'intégrer la même quantité de données sur les secteurs les plus proches du centre et ceux les plus éloignés
- Une solution technologique ZBR (Zone Bit Recording) sur disques durs récents
- ZBR consiste à définir différentes zones sur le disque en fonction de leur éloignement du centre du disque
- Le nombre de secteur par piste est ajusté dans chaque zone (plus on s'éloigne du centre plus les secteurs sont nombreux)



L'entrelacement

Problème :

- les secteurs physiques d'une piste sont adjacents les uns aux autres et ne sont pas séparés par beaucoup d'espace.
- La lecture consécutive de secteurs nécessite une certaine vitesse de la part du contrôleur de disque dur.
- Les plateaux n'arrêtent jamais de tourner, et dès que le contrôleur a fini de lire tout le secteur 1, il a peu de temps avant que le début du secteur 2 soit sous la tête
- Si le contrôleur est lent de cette manière, il doit attendre la rotations des plateaux avant que le début du secteur 2 arrive et qu'il puisse le lire.

L'entrelacement

Solution :

- Lorsque l'entrelacement est utilisé, les secteurs d'une piste sont logiquement renumérotés afin qu'ils ne correspondent pas à la séquence physique sur le disque.
- Le but de cette technique est d'arranger les secteurs de sorte que leur position sur la piste corresponde à la vitesse du contrôleur, pour éviter le besoin de "rotations" supplémentaires

L'adressage de l'espace disque

- Dans le format CHS, un secteur est identifié par son numéro **Cylindre/Tête/Secteur**.
- Au début, ces valeurs signifiaient l'emplacement physique du secteur sur le disque, plus tard, ces valeurs ne sont devenues que **des nombres logiques**.

Le format CHS a les restrictions suivantes :

numéro de cylindre	0 - 1023	(10 bits)
numéro de tête	0 - 254	(8 bits)
numéro de secteur	1 - 63	(6 bits)

- Avec 24 bits et une taille de secteur de 512 octets, il est possible d'adresser tous les secteurs d'un disque jusqu'à une taille de disque maximale de **8 Go**

Remarque : CHS est limité

L'adressage de l'espace disque

Remarque :

- Les plages de validité indiquées ne reflètent pas les faits physiques.
- Il n'existe pas de disques durs avec 128 plateaux (0-255 têtes).
- Ces valeurs maximales étaient autrefois utilisées par le BIOS pour adresser le disque dur.
- Le contrôleur de disque dur convertit ensuite les valeurs en caractéristiques réelles en interne.
- Il est nécessaire de prendre en charge l'adressage CHS pour des raisons de compatibilité, au moins pour le démarrage.

L'adressage de l'espace disque

Format LBA : (Logical Block Address)

- Cela signifie simplement que les secteurs d'un disque sont numérotés séquentiellement en commençant par le numéro **LBA 0**.
- Chaque secteur est identifié par son numéro LBA
- Cet adressage permet de désigner d'une façon unique un secteur d'un disque
- **L'adresse LBA** d'un secteur de données est simplement un numéro unique pris dans l'intervalle $[0 \dots N-1]$ où N est le nombre total de secteurs du support

Mappage CHS et LBA

- Il est possible de convertir le format **CHS en LBA**. Conceptuellement, les deux formes sont équivalentes. Un secteur C/H/S au format CHS porte le numéro LBA suivant :

$$\text{LBA} = C \times \text{Num_Head} \times \text{Num_Sec} + H \times \text{Num_Sec} + (S - 1)$$

- **Num_Sec** signifie le nombre (logique) de secteurs par piste
- **Num_Head** le nombre (logique) de têtes.
- Seuls ces deux paramètres de géométrie (logiques) du disque sont pertinents pour la conversion.

Mappage CHS et LBA

Les adresses LBA peuvent être mappé sur un tuple CHS :

$$C = \text{LBA} / \text{Num_Head} * \text{Num_Sec}$$

$$H = (\text{LBA} / \text{Num_Sec}) \bmod \text{Num_Head}$$

$$S = (\text{LBA} \bmod \text{Num_Sec}) + 1$$

- **Num_Sec** signifie le nombre (logique) de secteurs par piste
- **Num_Head** le nombre (logique) de têtes.

Remarque : certains SE et BIOS ont également été mis à jour pour prendre en charge le LBA, assurant ainsi une compatibilité avec les nouveaux disques durs.

Gestion des requêtes disque

Une requête spécifique :

- Type d'opération: entrée / sortie
- Adresse disque : numéro de bloc, traduit par le gestionnaire de disque en adresse disque composée
- Adresse mémoire: où copier ou vers ou copier
- Nombre octets à transférer

L'ordonnancement des requêtes

Objectif: minimiser les temps d'accès

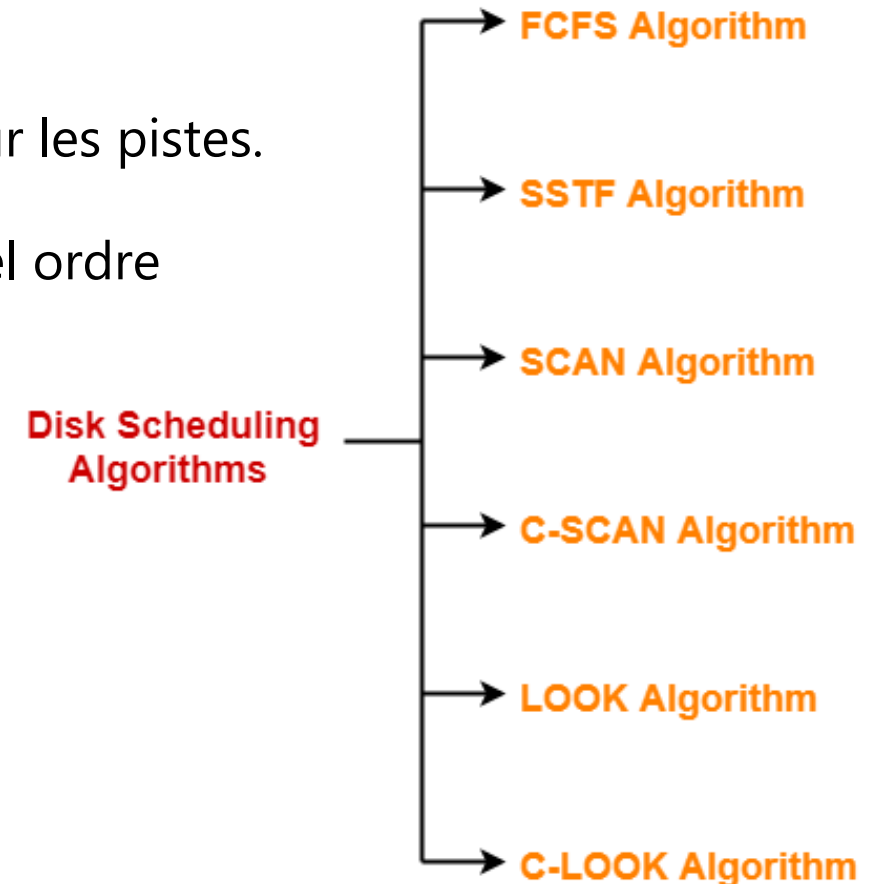
- Temps d'accès:
 1. Temps de positionnement du bras (seek time) = temps de déplacement de la tête de lecture/écriture sur la bonne piste (ou le bon cylindre)
 2. Temps de positionnement rotationnel = temps d'attente pour que le bloc désiré passe sous la tête
 3. Temps de transfert
- Le temps de positionnement (bras) est le plus important, donc il est celui que nous chercherons à minimiser

Les algorithmes d'ordonnancement

- Dans un système multiprogrammé avec mémoire virtuelle, il y a une file d'attente pour l'unité disque

Objectif : minimiser les déplacements de la tête de lecture sur les pistes.

- Etant donné la file de requêtes de lecture disque, dans quel ordre les exécuter?



Les algorithmes d'ordonnancement

- Nous étudierons différentes méthodes par rapport à une file d'attente arbitraire: 98, 183, 37, 122, 14, 124, 65, 67
- Chaque chiffre est un numéro séquentiel de cylindre
- Il faut aussi prendre en considération le cylindre de départ: 53
- **Hypothèse simpliste**: un déplacement coûte 1 unité de temps

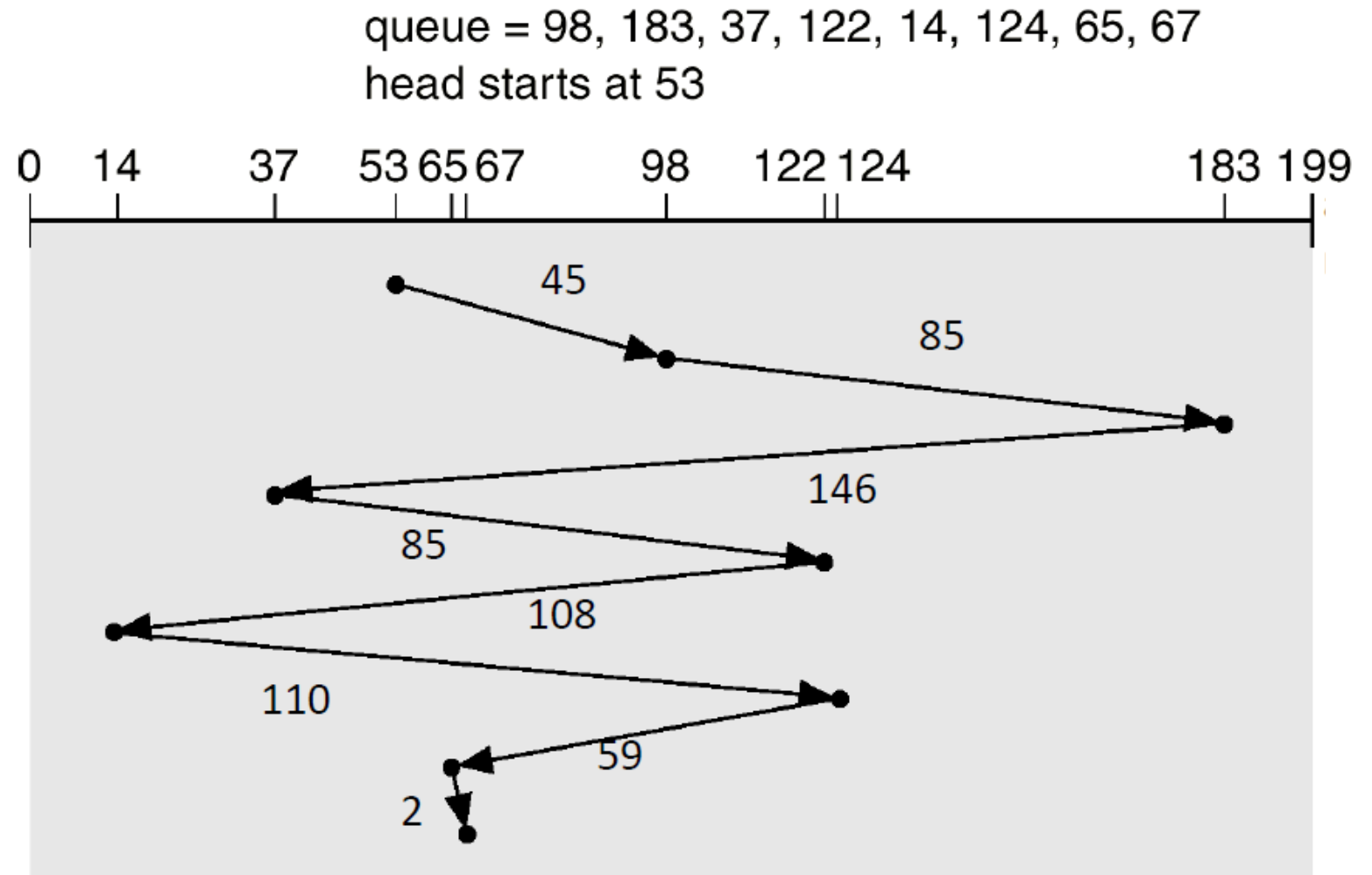
L'algorithme FIFO (FCFS)

Mouvement total :

$$\begin{aligned} & (98-53) + (183-98) + (183-37) \\ & + (122-37) + (122-14) + \\ & (124-14) + (124-65) + (67-65) \\ & = 640 \text{ cylindre/piste} \end{aligned}$$

En moyenne :

$$640/8 = 80$$



L'algorithme FIFO (FCFS)

- Avantages:

Ordonné selon l'ordre applicatif

- Inconvénient:

Aucune optimisation: Mouvements non optimisés de la tête de lecture

L'algorithme Shortest Seek Time First (SSTF)

- Sélectionner la requête la plus proche de la position courante de la tête (du cylindre courant)

Avantages:

- Minimise les temps de positionnement (clairement meilleur que le précédent)

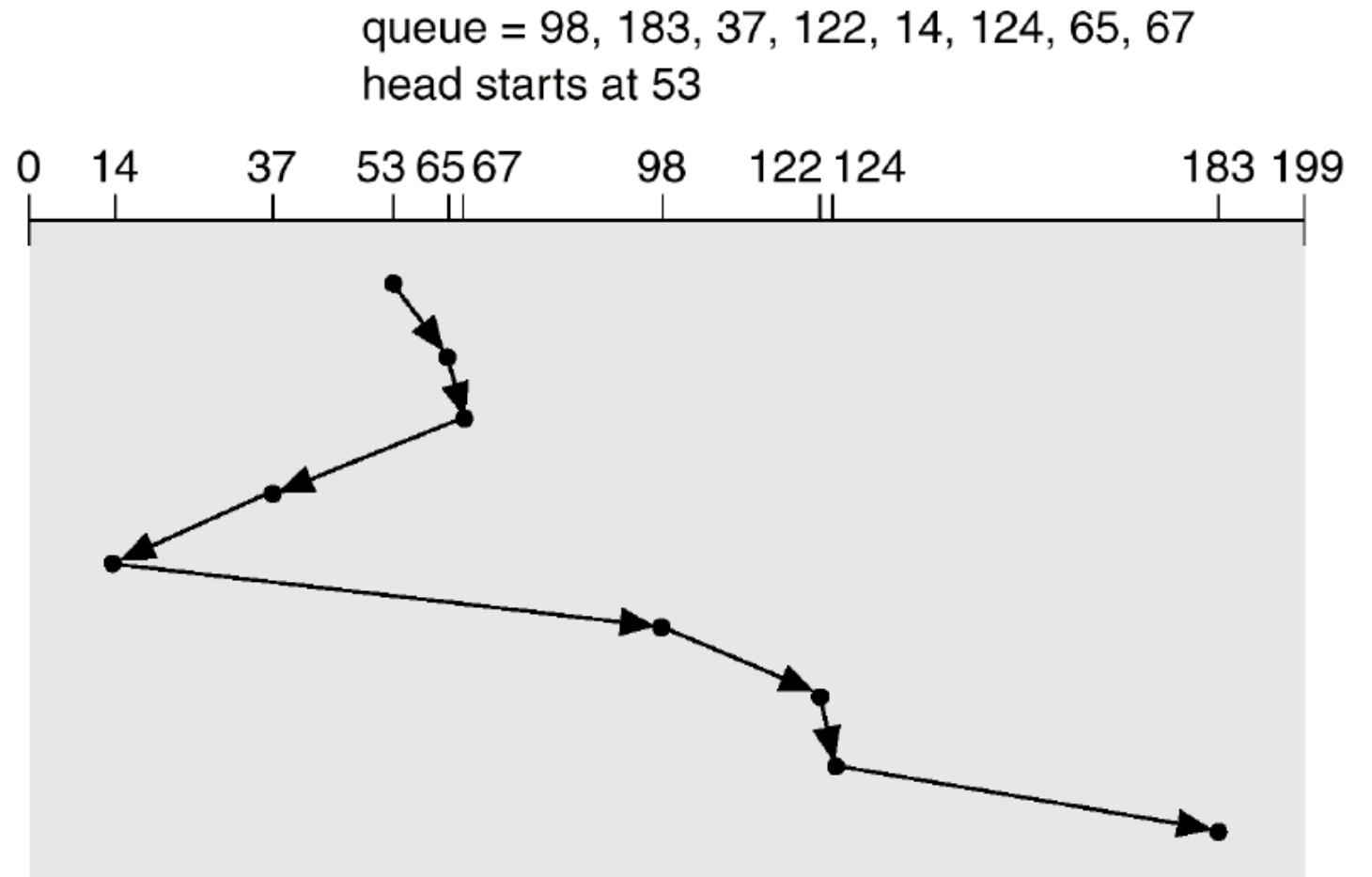
Inconvénient

- Peut causer la famine de certaines requêtes

L'algorithme Shortest Seek Time First (SSTF)

Mouvement total :
= 236 cylindre/piste

En moyenne :
 $236/8 = 29,5$



L'algorithme (SCAN)

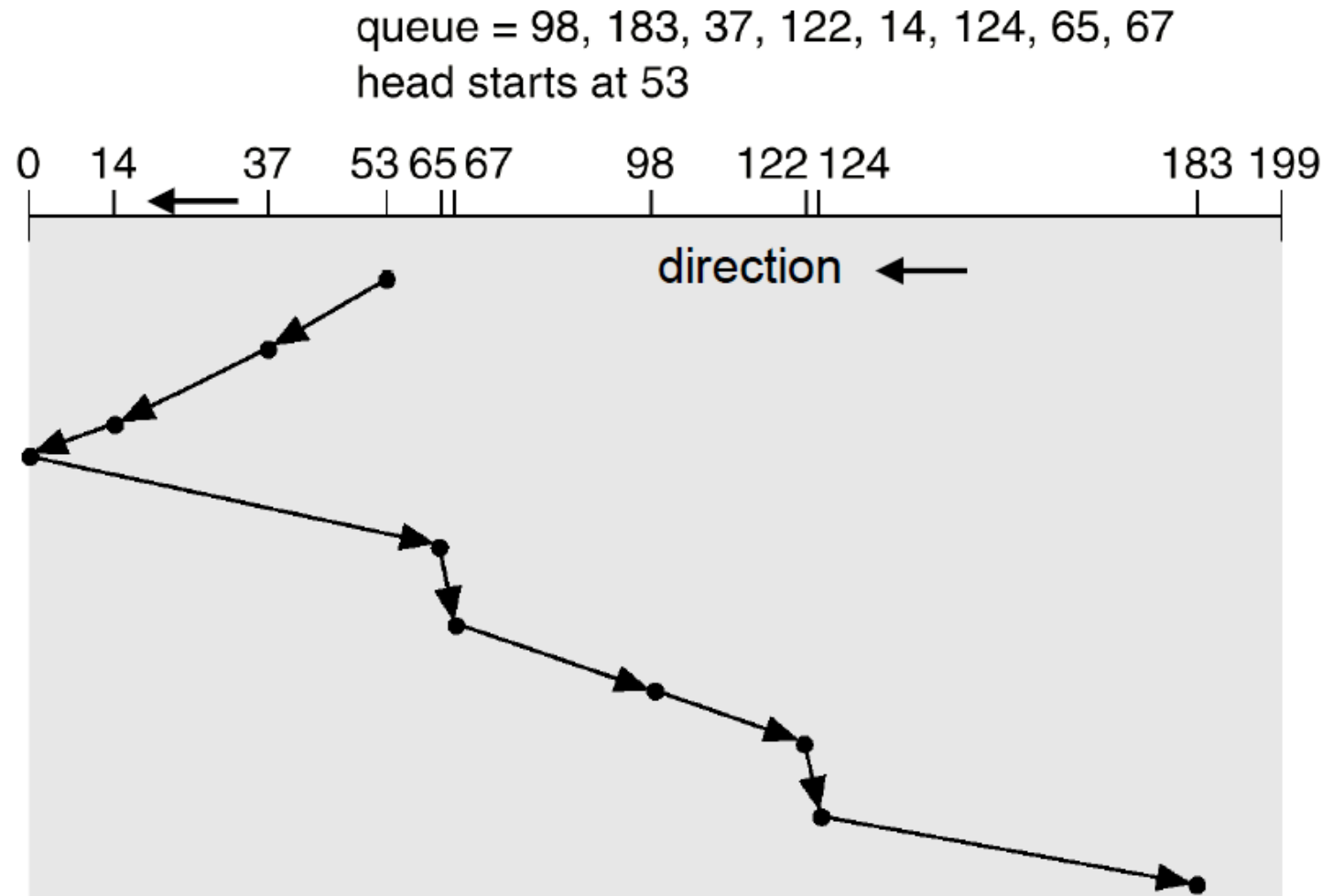
SCAN = balayage

- Appelé aussi **algorithme de l'ascenseur** (elevator)
- La tête balaye le disque dans une direction, puis dans la direction opposée, etc., en desservant les requêtes quand il passe sur le cylindre désiré

L'algorithme (SCAN)

Mouvement total :
= 236 cylindre/piste

En moyenne :
 $236/8 = 29,5$



L'algorithme (SCAN)

Avantages:

- Diminue le problème de famine

Inconvénient:

- Cela provoque le déplacement de la tête jusqu'à la fin du disque de cette manière, les demandes arrivant avant la position du bras obtiendraient un service immédiat, mais certaines autres demandes arrivant derrière la position du bras devront attendre que la demande se termine.

L'algorithme Circular SCAN (C-SCAN)

- Dans l'algorithme C-SCAN, le bras du disque se déplace dans une direction particulière pour répondre aux demandes jusqu'à ce qu'il atteigne le dernier cylindre
- Puis il saute au premier cylindre de la direction opposée sans répondre à aucune demande,
- Puis il se retourne et commence à se déplacer dans cette direction afin de traiter les demandes restantes.

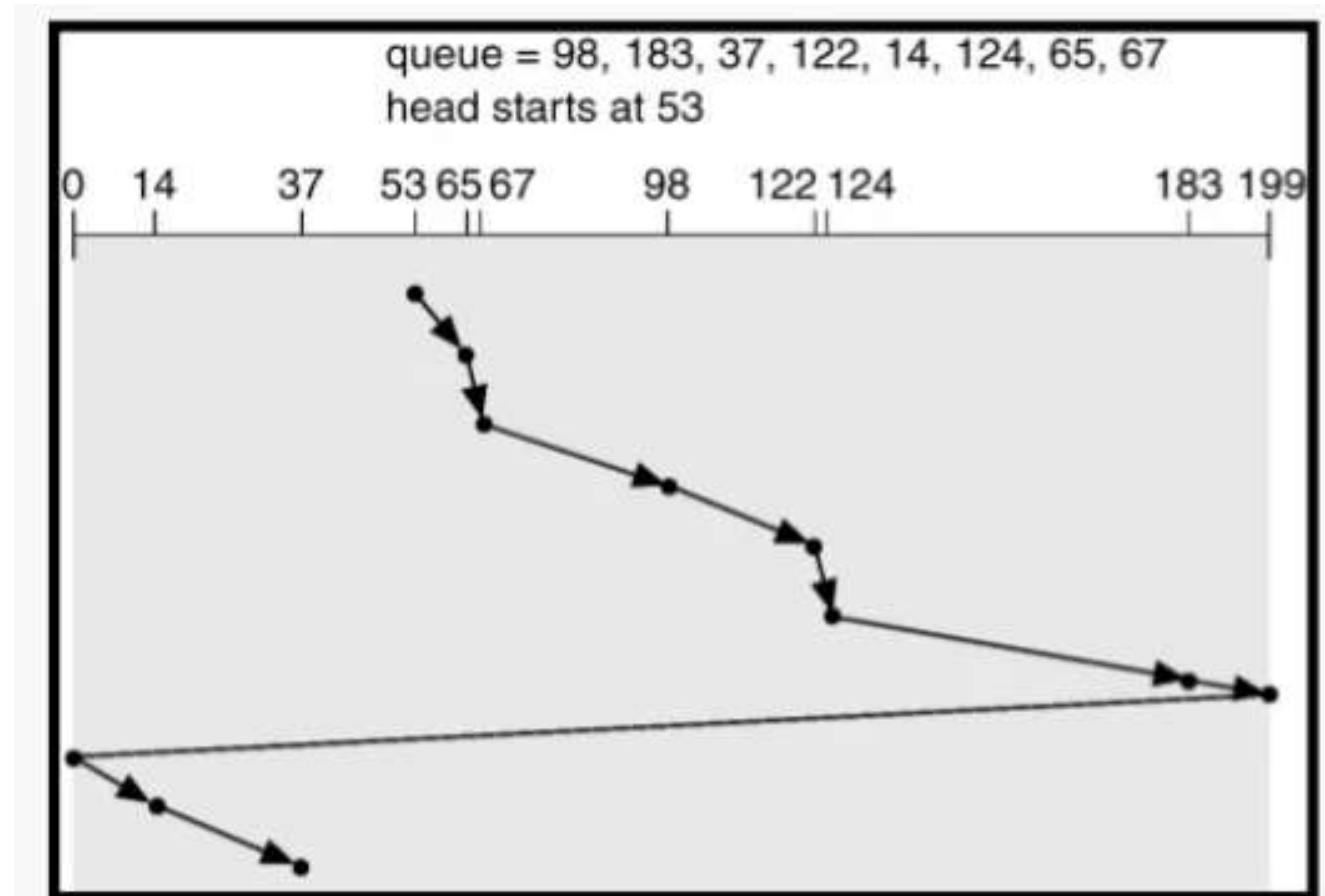
L'algorithme Circular SCAN (C-SCAN)

Mouvement total :

$$(199-53) + (37-0) \\ = 183 \text{ cylindre/piste}$$

En moyenne :

$$183/8 = 26$$



L'algorithme Circular SCAN (C-SCAN)

Avantage:

- Fournit un temps d'attente meilleur que SCAN

Inconvénient:

- Temps de déplacement de la tête sur le retour non exploité

L'algorithme (LOOK)

- Version améliorée de **SCAN**
- Lorsque la tête atteint la fin de l'extrémité, elle change de direction et revient à une requête, traitant toutes les requêtes intermédiaires.
- Contrairement à SCAN, au lieu d'aller à la dernière piste, dans **LOOK** cette tête va à la dernière requête puis change de direction.

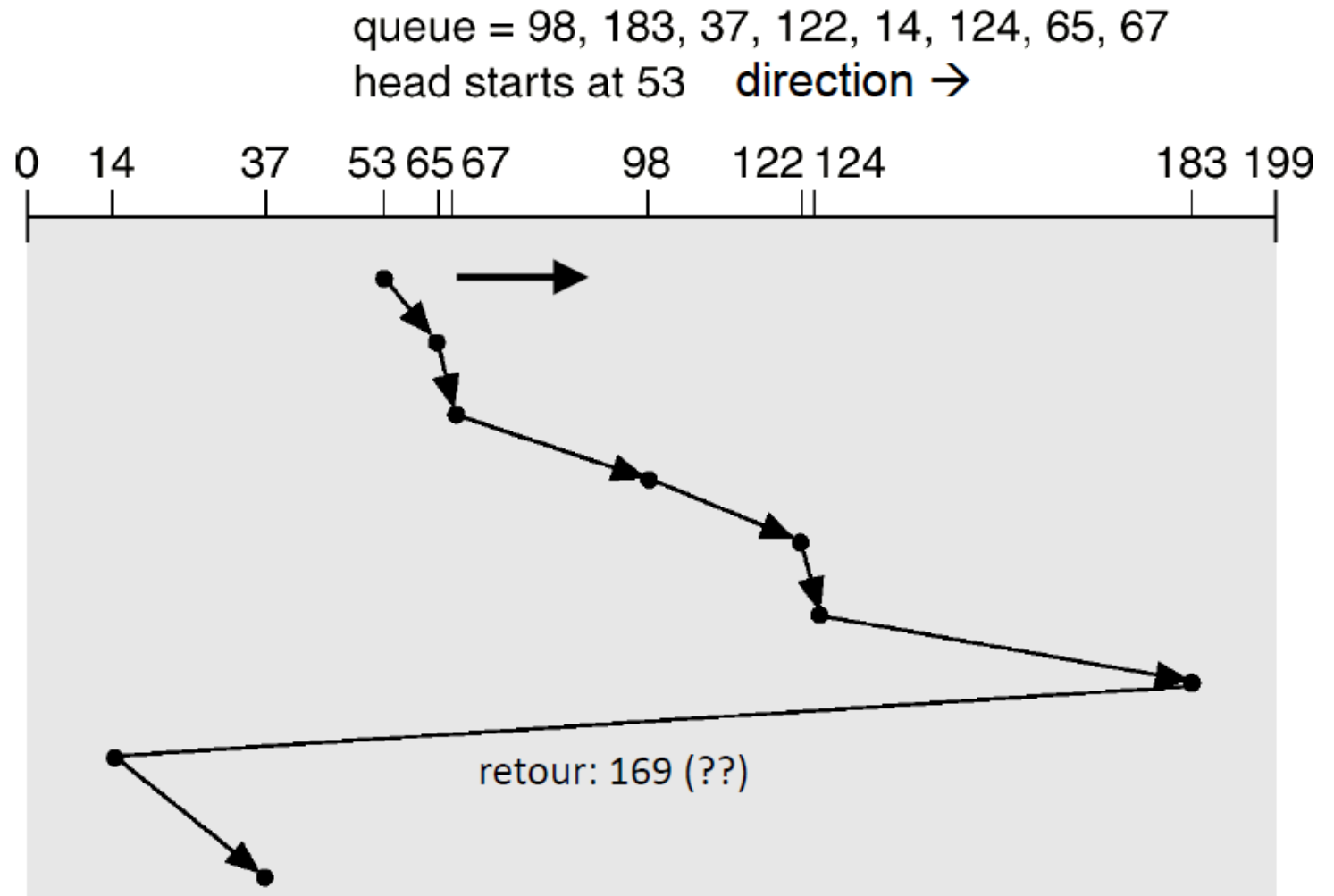
L'algorithme (C-LOOK)

- La même idée de **C-SCAN**, mais au lieu de retourner au cylindre 0, retourner au premier cylindre qui a une requête
- Commence à partir de la première requête du disque et se déplace vers une direction en traitant toutes les requêtes intermédiaires.
- Après avoir atteint la dernière requête à l'autre bout, la tête inverse sa direction.
- Il revient ensuite à la première demande à l'extrémité de départ sans traiter aucune demande entre les deux.

L'algorithme (C-LOOK)

Mouvement total :
= 153 cylindre/piste

En moyenne :
 $153/8 = 19,1$ sans considéré
le retour (169)
 $322/8 = 40,25$ avec retour
(169)



L'algorithme (N-STEP-SCAN)

- Il divise la file d'attente des demandes en **sous-files d'attente** de longueur N.
- Chaque sous-file d'attente est de longueur N où $N = 1, 2, 3..., n$.
- Lors du traitement d'une file d'attente, si de nouvelles requêtes arrivent, elles doivent être placées dans une autre file d'attente.

L'algorithme (N-STEP-SCAN)

- Soit les pistes demandées numérotées de 0 à 199 dans l'ordre 122, 90, 160, 24, 102, 89, 143, 18, 67. Pour $N = 2$, les pistes sont réparties dans les sous-files suivantes :

sous-file d'attente 1 = {122, 90}

sous-file d'attente 2 = {160, 24}

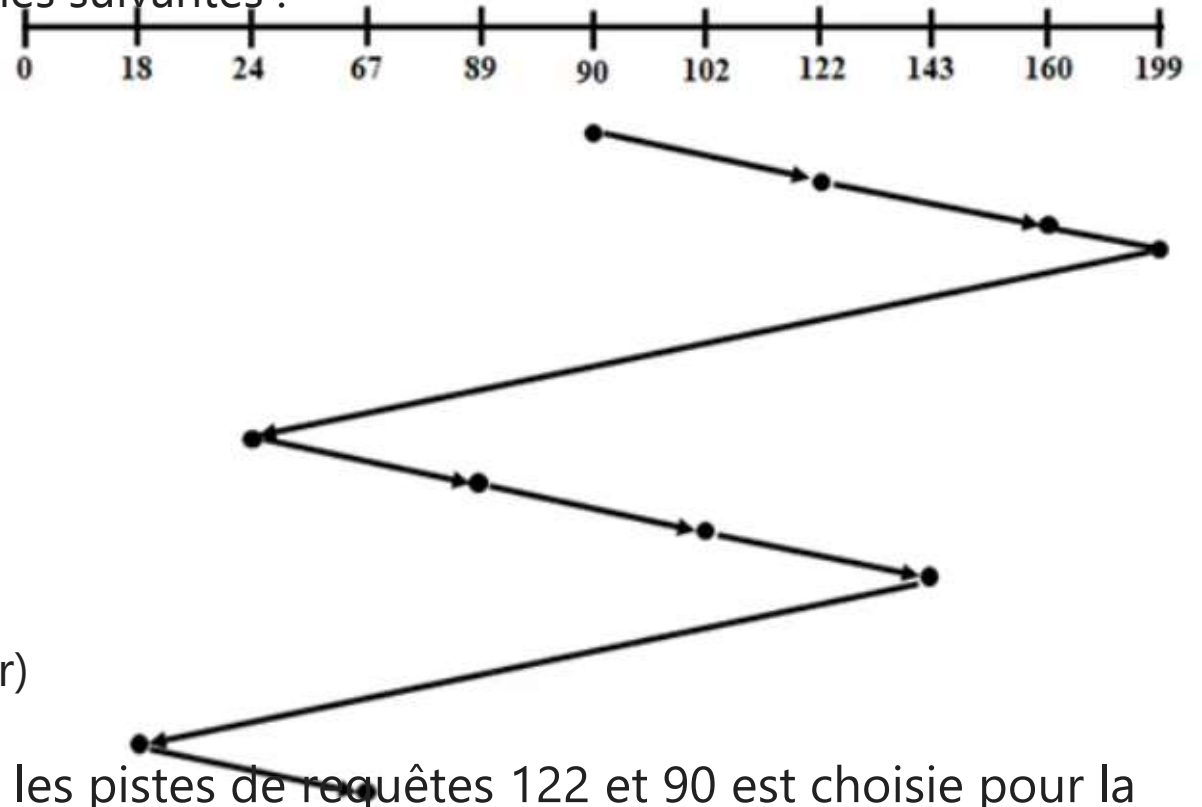
sous-file d'attente 3 = {102, 89}

sous-file d'attente 4 = {143, 18}

sous-file d'attente 5 = {67}

Mouvement total : 277 cylindre/Piste (sans retour)

Initialement, la sous-file d'attente 1 contenant les pistes de requêtes 122 et 90 est choisie pour la planification du disque. Comme la piste 90 est proche de 0, elle est traitée en premier.



L'algorithme Fixed period SCAN (FSCAN)

- Utilise deux sous-files d'attente .
- Lors de l' analyse , toutes les requêtes sont dans la première file d'attente et toutes les nouvelles requêtes sont placées dans le deuxième file d'attente.
- Ainsi ,les nouvelles demandes seront traités jusqu'à ce que les anciennes soient traitées
- Lorsque l'analyse se termine , le bras est dirigé vers les premières entrées de la file d' attente et recommence de zéro.