

HDD et SSD

Exercice 1. Questions de cours Les questions de cours sont à destinées à vous permettre de vérifier votre compréhension du cours. Elles sont à travailler à l'avance et ne seront pas traitées en TD ou TP.

1. Qu'est-ce qui définit l'emplacement d'un bloc sur un disque dur ?

Correction:

2. Quel est l'avantage de l'algorithme SSTF ? Et son inconvénient ?

Correction:

3. Quel est l'avantage des algorithmes de type look et scan ?

Correction:

4. Qu'est-ce que le wear leveling (WL) ? Quelles sont les principales techniques de WL ?

Correction:

5. Quel est l'avantage du wear leveling statique ?

Correction:

Exercice 2. Ordonnement d'accès aux blocs On considère un disque contenant 256 cylindres (numérotés de 0 à 255) et les demandes d'accès suivantes aux cylindres, avec leur date d'arrivée (les requêtes arrivées à la même date sont donnés dans l'ordre d'arrivée : on suppose que le contrôleur ne met à jour sa liste de requêtes que tous les 100 unités de temps). On suppose que la tête est en position 24 au début.

Date	Cylindre(s)
0	54, 13, 22, 188, 245, 98, 24, 167
100	67, 93, 12, 25, 250, 220, 200
200	94, 230, 97, 30, 20

On suppose qu'il faut une unité de temps pour parcourir un cylindre et, le cas échéant, lire les données.

1. Appliquez l'algorithme FCFS et calculez le temps de traitement total.

Correction: *Il s'agit de prendre les requêtes dans l'ordre. Le temps total est la somme des différences en valeur absolue :*

$$\begin{aligned}
 &30 + 41 + 9 + 166 + 57 + 147 + 74 + 143 \\
 &+ 100 + 26 + 81 + 13 + 225 + 30 + 20 \\
 &+ 106 + 136 + 133 + 67 + 10 = 1614
 \end{aligned}$$

Soit un total de 1614.

2. Appliquez l'algorithme SSTF et calculez le temps de traitement total.

Correction: *Tant qu'on n'a pas atteint 100 de déplacement, on ne considère pas la ligne suivante (c'est la difficulté pour cet algorithme et le suivant). En l'occurrence, on va faire : 24, 22, 13, 54, 98 (total = 96), 167 (total = 165) puis on aura en file les requêtes restantes (188 et 245) avec les nouvelles (67, 93, 12, 25, 250, 220, 200), ce qui donne 188, 200, 220 (total = 218) puis on aura les nouvelles requêtes, ce qui donne dans la file : 245, 67, 93, 12, 25, 250, 94, 230, 97, 30, 20 qui seront parcourues dans l'ordre : 230, 245, 250, 97, 94, 93, 67, 30, 25, 20, 12, soit un total de 486 (ce qui est 3 fois mieux que FCFS).*

3. Appliquez l'algorithme C-SCAN (en supposant qu'on est ascendant au début) et calculez le temps de traitement total.

Correction: *On va faire 24, 54, 98, 167 (total = 143) puis on reçoit la deuxième série, donc on va continuer 188, 200, 220, 245 (total = 221) puis on reçoit la troisième série (il reste donc 13, 22, 67, 93, 12, 25, 250, 94, 230, 97, 30, 20) donc on fait 250 (dommage pour le 230 qui vient de passer) et on continue jusqu'à 255 puis on reprend au début (on a déjà parcouru 231) et on remonte jusqu'à 230. Total 461 qui est finalement mieux que le SSTF !*

Note : Mais en fait, très probablement, à t=200, lorsque la requête 230 arrive, la tête est en progression du côté du 224 donc elle peut attraper le 230. Dans ce cas, le temps total descend à

328. Mais on ne connaît pas les détails techniques du disque et il est donc impossible de le prédire ici.

Exercice 3. Ordonnancement d'accès aux blocs On considère un disque dur comportant 200 cylindres et dont la tête de lecture est actuellement à la position 34. Le contrôleur met à jour sa liste de requêtes toutes les 100 unités de temps. On donne ci-dessous la liste des requêtes données au contrôleur :

Date	Cylindre(s)
0	78, 134, 96, 189, 41
100	18, 151
200	94, 154, 41

1. Appliquez l'algorithme FCFS et calculez le temps de traitement total.

Correction: On prend ici les requêtes dans l'ordre, le temps total est la somme des écarts entre les valeurs consécutives. $44 + 56 + 38 + 93 + 148 + 23 + 133 + 57 + 60 + 111 = 763$.

2. Appliquez l'algorithme SSTF et calculez le temps de traitement total.

Correction: Attention de bien repérer le moment où les nouvelles requêtes arrivent. On obtiens un total de 462 unités de temps.

Temps	Position	Requête	Durée
0	34	41	+7
7	41	78	+37
44	78	96	+18
62	96	134	+38
100	134	151	+17
117	151	189	+38
155	189	18	+171
326	18	41	+23
349	41	94	+53
402	94	154	+60
462			

Exercice 4. Ordonnancement d'accès aux blocs On considère un disque composé de 256 cylindres (numérotés de 0 à 255) recevant des requêtes d'accès à des blocs situés sur des cylindres différents. On s'intéresse ici uniquement aux cylindres. On suppose que le déplacement de la tête d'un cylindre à l'autre prend 1 unité de temps. On suppose que la vitesse de rotation du cylindre est de 10 unités de temps et, par conséquent, que le traitement d'une requête prend toujours 10 unités de temps une fois la tête positionnée.

La table ci-dessous décrit les arrivées de requêtes sur le contrôleur de disque :

Date	Cylindre(s)
0	167, 21
53	213, 98
135	87
215	4, 8, 131
523	213, 81

Initialement, la tête est positionnée sur le cylindre 37 et il n'y a aucune requête dans la file d'attente.

Important : une requête de déplacement du bras ne peut pas être interrompue : si une nouvelle requête plus prioritaire arrive pendant que le bras se déplace pour atteindre la prochaine requête, elle sera traitée ultérieurement.

1. Décrivez l'exécution si le contrôleur utilise un algorithme FCFS et donnez le temps d'exécution total.

Correction: Dans le tableau suivant, les dates sont indiquées en fin de lecture du cylindre :

Date	0	140	296	498	623	644	737	751	884	976	1118
Cyl.	37	167	21	213	98	87	4	8	131	213	81

2. Décrivez l'exécution si le contrôleur utilise un algorithme SSTF et donnez le temps d'exécution total.

Correction:

Date	0	26	182	238	330	373	394	483	497	523	610	752
Cyl.	37	21	167	213	131	98	87	8	4	4	81	213

Au temps 26, nous n'avons pas reçu les suivants donc nous allons au 167.

Au temps 182, nous avons reçu 87, 98 et 213. SSTF choisit 213 qui est la plus proche.

Au temps 238, nous avons reçu 4, 8, 87, 98, 131. SSTF choisit 131 qui est la plus proche, puis les suivantes dans l'ordre décroissant, qui seront toutes traitées avant la date 523 de la nouvelle série de requêtes

Au temps 497, nous n'avons plus rien en stock, donc nous sommes attendons le temps 523, où nous traitons 81 puis 213.

3. Décrivez l'exécution si le contrôleur utilise un algorithme C-LOOK et donnez le temps d'exécution total. On suppose que la tête de lecture est montante dans l'état initial.

Correction:

Date	0	140	196	227	303	324	367	497	511	523	606	748
Cyl.	37	167	213	21	87	98	131	4	8	8	81	213

Au départ, on va vers 167. À 140, on a dans la file 21, 87, 98 et 213 donc on continue vers 213.

À 196, on n'a plus rien en montant, donc on repart d'en bas (21). 213 est à 42 du bord, donc il faut encore se déplacer de $(42-21)=21$ avant de lire

À 227, on a dans la file 4, 8 (dépassés), 87, 98, 131. On continue en montant jusqu'à 131.

À 367, il reste 4 et 8 dans la file. Nous n'avons pas encore reçu les requêtes 213 et 81. Donc nous allons jusqu'à 4, c'est-à-dire par symétrie $255-4=251$. Il faut donc avancer de 120 pas.

Au temps 511, nous devons attendre la prochaine requête, que nous traitons dans l'ordre montant.

Exercice 5. Disque SSD On considère un disque SSD de 8 Mo répartis en blocs de 512 Ko numérotés de 0 à F (en hexadécimal). La table suivante donne les numéros de blocs physiques, les blocs logiques qu'ils contiennent (ou L lorsque le bloc est libre), la date de dernière écriture (exprimée en unités de temps depuis le début de la vie du disque) et l'usure (en nombre d'utilisation depuis le début d'utilisation du disque): actuellement, il y a 10 blocs logiques (0 à 9) qui sont stockés sur ce disque.

phys.:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log.:	1	4	2	6	0	L	9	3	L	L	5	7	8	L	L	L
date:	53	41	18	99	72	87	55	31	80	58	64	67	91	61	0	0
usure:	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0

Nous sommes à l'unité de temps 100 et on suppose que l'UC fait, dans l'ordre, des requêtes d'écriture sur les pages situées sur les blocs logiques suivants (1 requête par unité de temps):

A 1 7 1 2 B A

1. Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique de Wear Levelling dynamique. Expliquez clairement ce qu'il se passe à chaque pas de temps de l'exécution.

Correction:

100 (A) Nouveau bloc → prendre le segment libre le moins usé: c'est E

101 (1) Remplacer 1 → prendre le segment libre le moins usé: c'est F; marquer 0 libre

102 (7) Remplacer 7 → prendre le segment libre le moins usé: c'est 0 maintenant qu'il a été libéré; marquer B libre

103 (1) Remplacer 1 → prendre le segment libre le moins usé: c'est D; marquer F libre à nouveau

104 (2) Remplacer 2 → prendre le segment libre le moins usé: c'est F; marquer 2 libre

105 (B) Nouveau bloc → prendre le segment libre le moins usé: c'est 5

106 (A) Remplacer A → prendre le segment libre le moins usé: c'est 2; marquer E libre

phys.:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log.:	7	4	A	6	0	B	9	3	L	L	5	L	8	1	L	2
date:	102	41	106	99	72	105	55	31	80	58	64	67	91	103	100	104
usure:	3	8	8	3	6	6	8	6	7	9	6	8	8	4	1	2

2. Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique de Wear Leveling statique avec un délai d'ancienneté fixé à 10 unités de temps. Expliquez clairement ce qu'il se passe à chaque pas de temps de l'exécution.

Correction:

100 (A) On cherche le bloc logique le moins utilisé: c'est E. Il est libre: on l'utilise.

101 (1) On cherche le bloc logique le moins utilisé: c'est F. Il est libre: on l'utilise. On marque 0 libre.

- 102 (7) On cherche le bloc logique le moins utilisé : c'est E. Il n'est pas libre et il a été utilisé récemment. Idem pour F. Le prochain moins utilisé est 0 qui est libre : on l'utilise et on marque B libre.
- 103 (1) On cherche le bloc logique le moins utilisé. On élimine E, F, 0 et 3 car ils sont non-libres et trop récents. On arrive à D qui est libre et on l'utilise. On marque F libre.
- 104 (2) On cherche le bloc logique le moins utilisé : c'est F et il est libre. On l'utilise et on marque 2 libre.
- 105 (B) On cherche le bloc logique le moins utilisé. On élimine E, F, 0, 3 et D qui sont non-libres et trop récents. On arrive à 5 qui est libre et on l'utilise.
- 106 (A) On cherche le bloc logique le moins utilisé. On élimine E, F, 0, 3, D et on arrive à 4 qu'on utilise et on marque E libre. Mais il faut alors déplacer le bloc 0.
- 106b (0) On cherche le bloc logique le moins utilisé : c'est E et il est libre. On l'utilise.

phys. :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log. :	7	4	L	6	A	B	9	3	L	L	5	L	8	1	0	2
date :	102	41	18	99	106	105	55	31	80	58	64	67	91	103	106	104
usure :	3	8	7	3	7	6	8	6	7	9	6	8	8	4	2	2

Exercice 6. Disques SSD On considère un disque SSD de 8 Mo répartis en blocs de 512 Ko numérotés de 0 à F (en hexadécimal). La table suivante donne les numéros de blocs physiques, les blocs logiques qu'ils contiennent (ou L lorsque le bloc est libre), la date de dernière modification (exprimée en unités de temps depuis le début de la vie du disque) et l'usure (en nombre d'utilisation depuis le début d'utilisation du disque) est donnée par la table suivante :

phys. :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log. :	8	3	2	1	0	L	B	6	L	9	5	7	4	L	L	L
date :	121	112	123	107	110	100	130	158	120	151	132	128	117	050	0	0
usure :	7	8	7	9	6	5	8	6	7	9	6	8	8	3	0	0

Nous sommes à l'unité de temps 182 et on suppose que l'UC fait, dans l'ordre, des requêtes d'écriture sur les pages situées sur les blocs logiques suivants (1 requête par unité de temps) :

A 7 8 7 1 0 A B

1. Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique qui ne fait aucun Wear Levelling. Expliquez vos choix (1 point)

Correction: Si on ne fait aucun WL, les blocs ne sont pas déplacés. Les blocs logiques 0, 1, 7, 8 et B sont donc modifiés dans les blocs physiques correspondants (4, 3, B, 0 et 6), dont l'usure augmente de 1 (2 pour le bloc B contenant le bloc logique 7) et l'ancienneté modifiée en fonction de l'usage (respectivement 187, 186, 185, 184 et 189). Enfin, pour le bloc logique A, il faut choisir un bloc libre. Il n'y a aucune consigne dans l'énoncé quant à la table d'association entre blocs physiques et blocs logiques, donc A peut être placé n'importe où (nous avons choisi le premier bloc libre, mais c'est discutable).

phys. :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log. :	8	3	2	1	0	A	B	6	L	9	5	7	4	L	L	L
date :	184	112	123	188	187	188	189	158	120	151	132	185	117	050	0	0
usure :	8	8	7	10	7	7	9	6	7	9	6	10	8	3	0	0

2. Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique qui fait du Wear Levelling dynamique. Expliquez vos choix (2 points)

Correction: En WL dynamique, on déplace les blocs écrits vers le bloc libre le moins utilisé, ce qui donne :

182 Pour le bloc logique A, on prend le bloc libre le moins utilisé (c'est E)

183 Pour le bloc logique 7, on le déplace vers F (usure=0) donc B est libre

184 Pour le bloc 8, on le déplace de 0 vers D (usure=3)

185 Pour le bloc 7, on le déplace de F vers 5 (usure=5)

186 Pour le bloc 1, on le déplace de 3 vers F qui est redevenu libre (usure=1)

187 Pour le bloc 0, on le déplace de 4 vers 0 (usure = 7)

188 Pour le bloc A, on le déplace de E vers 4 (usure = 6)

189 Pour le bloc B, on le déplace de 6 vers E (usure = 1)

phys. :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log. :	0	3	2	L	A	7	L	6	L	9	5	L	4	8	B	1
date :	187	112	123	107	110	185	130	158	120	151	132	128	117	184	189	186
usure :	8	8	7	9	7	6	8	6	7	9	6	8	8	4	2	2

3. Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique de Wear Levelling *statique* avec un délai d'ancienneté fixé à 10 unités de temps. Expliquez clairement ce qu'il se passe à chaque pas de temps de l'exécution. (2 points)

NB : En cas de doute, toutes choses étant égales par ailleurs, vous choisirez le premier bloc dans l'ordre numérique.

Correction:

182 Pour le bloc logique A, on cherche le bloc physique le moins utilisé (c'est E). Il est libre, on l'utilise.

183 Pour le bloc logique 7, même principe. On place sur donc en F et on marque B comme libre.

184 Pour le bloc 8, le moins utilisé est à nouveau E, mais il a été utilisé il y a moins de 10 pas de temps. Idem pour F. Donc on se place en D (qui est libre) et on marque 0 comme libre.

185 Pour le bloc 7 à nouveau, les plus anciens (D, E et F) sont non-libres et récents, donc on se place en 5 (qui est le suivant sur la liste, et qui est libre). On marque F comme libre.

186 Pour le bloc 1, on utilise F libre et on marque 3 comme libre.

187 Pour le bloc 0, blocs D, E et F sont peu utilisés mais récents. Les candidats (usure=6) sont donc 4, 5, 7 et A. Aucun n'est libre mais 5 est récent, donc le premier non-libre et non-récent est 4. Il contient justement le bloc 0 (donc on n'a rien à déplacer).

188 Pour le bloc A, on est dans la même situation que précédemment (4, 5, D, E et F peu utilisés mais trop récents), il ne reste que 7 et A, qui ne sont pas libre. On prend le premier (c'est le bloc 7) pour y mettre A et peut marquer E comme libre. Il faut maintenant placer le bloc 6 ailleurs

188bis Pour le bloc 6, on a E qui le moins utilisé des blocs libres.

189 Pour le bloc B, le moins utilisé non-récent est donc le bloc A. On y place notre B, on marque 6 comme libre et on déplace 5 dans le bloc libre le moins utilisé, qui est 0

phys. :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log. :	5	3	2	L	0	7	L	A	L	9	B	L	4	8	6	1
date :	189	112	123	107	187	185	130	188	120	151	189	128	117	184	188	186
usure :	8	8	7	9	7	6	8	7	7	9	7	8	8	4	2	2

Exercice 7. Disque SSD On considère un disque SSD de 4 Mo comprenant 8 blocs de 512Ko, dont la table LBA est donnée ci-dessous :

phys.	0	1	2	3	4	5	6	7
log.	3	L	1	4	0	L	2	L
date	769	534	681	210	721	754	589	345
usure	135	97	111	54	154	143	81	67

À partir de la 770ème écriture, le disque reçoit les requêtes d'écriture sur les blocs suivants :

5 1 2 1 3 1

1. Quelle est l'usure du secteur 2 à la fin de l'exécution si on utilise une allocation simple, sans gestion de l'usure ?

Correction: Il y a 3 requêtes d'écriture sur le bloc 1 contenu dans le secteur 2, donc l'usure à la fin de l'exécution est $111 + 3 = 114$.

2. Quel bloc contient le secteur 0 à la fin de l'exécution si on utilise un algorithme de gestion de l'usure dynamique ?

Correction: À la fin de l'exécution, le secteur 0 contiendra le bloc 1.

Nous pouvons dérouler l'exécution en entier pour bien comprendre ce qu'il se passe :

t=770 bloc 5 placé dans le secteur 7

t=771 bloc 1 placé dans le secteur 1, secteur 2 libre

t=772 bloc 2 placé dans le secteur 2, secteur 6 libre

t=773 bloc 1 placé dans le secteur 6, secteur 1 libre

t=774 bloc 3 placé dans le secteur 1, secteur 0 libre

t=775 bloc 1 placé dans le secteur 0, secteur 6 libre

Ce qui donne à la fin :

phys.	0	1	2	3	4	5	6	7
log.	1	3	2	4	0	L	L	5
usure	136	99	112	54	154	143	82	68

Mais plus simplement, il suffit de remarquer que le bloc 3, contenu dans le secteur 0, est modifié à la 774e écriture sur le disque. Il sera donc déplacé, rendant le secteur 0 libre. La question est de savoir si le bloc 1 sera déplacé dans ce secteur ou dans un autre secteur moins usé.

On peut noter qu'à la 770e écriture (bloc 5), on perd un secteur libre. Le jeu se fait donc systématiquement entre deux secteurs libres : le secteur 5 (usure 143) et un autre secteur, qui va changer à chaque exécution, mais donc l'usure sera forcément plus faible sauf s'il s'agit du secteur 4. Mais cela ne peut pas être le secteur 4 car il contient le bloc 0 qui n'est jamais écrit. Donc à la 775e écriture, le bloc libre le moins usé sera bien le secteur 0, et c'est là que le bloc 1 sera placé.

3. Quelle est l'usure du secteur 3 à la fin de l'exécution si on utilise un algorithme de gestion de l'usure statique avec une limite d'ancienneté $t_1 = 1$: seuls les blocs déplacés au tour précédent sont protégés ? Justifiez votre réponse.

Correction: Le secteur 3 aura une usure de 57. Pour le comprendre, nous pouvons simplement dérouler l'exécution :

t=770 bloc 5 placé dans le secteur 3, bloc 4 déplacé dans secteur 7 (libre)

t=771 bloc 1 déplacé dans le secteur 6, secteur 2 libre, bloc 2 (ex-secteur 6) déplacé dans secteur 1 (libre)

t=772 bloc 2 déplacé dans le secteur 3, secteur 1 libre, bloc 5 (ex-secteur 3) déplacé dans secteur 7, bloc 4 déplacé dans le secteur 1 (libre)

t=773 bloc 1 reste en place (secteur le moins usé et non protégé)

t=774 bloc 3 déplacé secteur 3, secteur 0 libre, bloc 2 (ex-secteur 3) déplacé dans secteur 7, bloc 5 déplacé dans secteur 1, bloc 4 déplacé dans secteur 2 (libre)

t=775 bloc 1 reste en place

Ce qui donne :

phys.	0	1	2	3	4	5	6	7
log.	L	5	4	3	0	L	1	2
date	769	774	774	774	721	754	773	774
usure	135	100	112	57	154	143	83	70

Mais plus simplement, il suffit de remarquer que le secteur 3 est le moins usé. C'est donc lui qui sera toujours choisi. Il n'a droit qu'à un cycle de répit. Il sera donc choisi aux écritures 770, 772 et 774. Son usure sera alors $54+3 = 57$.