

## Série d'exercices 8 : Entrées et sorties

- 8.1 Un numériseur traite une page de format lettre en couleur, avec une résolution de 600 pixels par pouce, en 5 secondes. L'image obtenue est compressée par un facteur 10 par rapport à l'image brute. Est-il possible de numériser un document et d'envoyer efficacement le résultat par réseau sur un réseau local à 100Mb/s?
- Le numériseur produit  $8.5 \text{ pouces} \times 600 \text{ p/pouce} \times 11 \text{ pouces} \times 600 \text{ p/pouce} \times 3 \text{ octets par pixel} / 10 / 5s = 2019600 \text{ octet/s}$  alors que la bande passante disponible est d'environ 12.5Mo/s. Cela ne pose donc pas de problème.
- 8.2 Un système utilise des entrées sorties calquées sur la mémoire et contient deux bus, un pour la mémoire et l'autre pour les entrées sorties. Le système envoie chaque accès mémoire sur le bus mémoire en premier et, en cas d'échec, l'envoi sur le bus des dispositifs d'entrée sortie. On vous propose de changer cette organisation afin de plutôt envoyer la requête sur les deux bus simultanément. Serait-ce avantageux?
- L'idée est intéressante à première vue mais en fait le bus d'entrée sortie est normalement beaucoup plus lent et deviendrait le goulot d'étranglement du système. En effet, il faudrait attendre après la demande d'accès courante avant de faire la demande d'accès suivante sur ce bus plus lent.
- 8.3 Un ordinateur peut accéder un mot en mémoire en 10ns. Lors d'une interruption, il doit sauver 32 registres d'usage général, le compteur de programme et le registre de statut. Quel est le nombre maximal d'interruptions par seconde que ce système pourrait théoriquement soutenir?
- Avec 34 registres à sauver puis à recharger à chaque interruption, il lui faut au moins  $34 \times 2 \times 10\text{ns} = 680\text{ns}$ . Il est donc limité à un maximum de  $1s / 680\text{ns/Int.} = 1.47\text{MHz}$ .
- 8.4 Un gestionnaire d'interruption doit envoyer le prochain caractère à l'imprimante et réactiver les interruptions. Est-ce qu'un ordre est préférable à un autre?
- En envoyant le caractère immédiatement, cela occupe l'imprimante plus vite par rapport à si les interruptions étaient réactivées et une autre interruption en attente s'exécutait immédiatement. Par contre, le fait d'envoyer le caractère avant de réactiver les interruptions augmente légèrement la latence d'interruption.
- 8.5 Une imprimante peut traiter 6 pages de 50 lignes de 80 caractères par minute. L'envoi de chaque caractère sur le port parallèle par le CPU est très court. Est-ce efficace de faire ces transferts avec une interruption pour chaque caractère, considérant que l'interruption prend 50us?

Il y aura donc  $6 \text{ p/m} * 50 \text{ l/p} * 80 \text{ c/l} / 60 \text{ s/m} = 400 \text{ c/s}$ . Le temps passé en interruption sera donc de  $400 * 50\mu\text{s} = 20\text{ms}$  à chaque seconde, soit 2% du temps.

- 8.6 Dans quelle couche du système est-ce que chacune de ces opérations est normalement effectuée : a) calcul de la piste/secteur/tête d'une lecture, b) écriture d'une commande dans le registre d'un dispositif d'entrée sortie, c) vérification des droits d'accès pour un dispositif, d) conversion de données binaires en codes ASCII pour impression?

Pour a), ceci est effectué dans le pilote d'interface, possiblement dans une portion partagée entre plusieurs gestionnaires, si plusieurs types de disques ont une manière semblable de faire ce calcul. Le cas de b) est assurément dans le pilote d'interface, alors que c) est au niveau du système de fichiers virtuel dans le système d'exploitation. Finalement, d) est normalement effectué en mode usager.

- 8.7 L'envoi d'un paquet suit le chemin suivant sur un système : appel système, copie en mémoire système, copie vers la carte réseau, envoi à 10Mb/s, réception 1us après l'envoi, génération d'une interruption lorsque le paquet est complètement reçu, copie en mémoire système et copie en mémoire usager. Si chaque paquet fait 1024 octets, la copie de chaque octet prend 1us et l'interruption prend 1ms, quel est le débit maximal d'envoi entre deux processus si on attend l'accusé de réception (très petit) avant d'envoyer le paquet suivant?

L'envoi demande deux copies ( $2 * 1024 * 1\mu\text{s} = 2.048\text{ms}$ ), et l'envoi à 10Mb/s ( $1024 \text{ octets} * 8 \text{ bits / octet} / 10\text{Mbit/s} = .8192\text{ms}$ ). Le transit est 1us. La réception prend une interruption (1ms) et deux copies (2.048ms). Le total est donc  $2.048 + .8192 + .001 + 1 + 2.048 = 5.9162\text{ms}$ . L'accusé de réception prend environ le temps de l'interruption, le reste étant négligeable étant donné sa petite taille. Ceci donne donc  $1024 \text{ octets} / (5.9162\text{ms} + 1\text{ms}) = 148058 \text{ o/s}$ .

- 8.8 De combien devrait-on décaler le début de chaque piste si la tête se déplace d'une piste en 1ms, le disque tourne à 7200rpm et chaque piste contient 200 secteurs de 512 octets?

En 1ms, le disque peut faire :  $7200 \text{ t/m} / 60\text{s/m} * 1\text{ms} = .12 \text{ t}$ , soit  $.12 * 200 = 24$  secteurs. En sautant environ 24 secteurs, le disque devrait pouvoir lire séquentiellement, sans perte de temps, deux pistes consécutives.

- 8.9 Un disque a une probabilité p de faire défaut pendant 1 heure. Quelle est la probabilité qu'une unité de disque RAID, avec k disques et pouvant tolérer la défaillance d'un disque, fasse défaut pendant une heure donnée?

La probabilité que tous les disques soient opérationnels est  $(1 - p)^k$  et la probabilité qu'un des disques soit brisé est  $k p(1 - p)^{k-1}$ . La probabilité de défaillance sera donc  $1 - (1 - p)^k - k p(1 - p)^{k-1}$ .

- 8.10 Les requêtes arrivent pour accéder les cylindres 10, 22, 20, 2, 40, 6 et 38. Si la tête prend 6ms par cylindre pour se déplacer, quel est le temps requis selon l'ordonnancement utilisé avec une position initiale à 20 : a) FIFO, b) le plus près, c) ascenseur qui est initialement en phase montante?

En a), nous aurons 20, 10, 22, 20, 2, 40, 6 et 38, soit des déplacements de  $10 + 12 + 2 + 18 + 38 + 34 + 32 = 146 * 6\text{ms} = 876\text{ms}$ . Avec b), cela devient : 20, 20, 22, 10, 6, 2, 38, 40, soit un déplacement de  $0 + 2 + 12 + 4 + 4 + 36 + 2 = 60 * 6\text{ms} = 360\text{ms}$ . Finalement, avec c) on obtient : 20, 20, 22, 38, 40, 10, 6, 2 soit un déplacement de  $0 + 2 + 16 + 2 + 30 + 4 + 4 = 58 * 6\text{ms} = 348\text{ms}$ .

- 8.11 Le professeur louange la performance de l'algorithme de l'ascenseur utilisé dans le système d'exploitation Linux. Un étudiant fait un test avec un programme qui lit aléatoirement 10000 blocs bien répartis sur le disque. Il ne note aucune différence par rapport au même test fait avec un ordonnanceur de disque trivial, qui sert les requêtes dans leur ordre d'arrivée. Comment expliqueriez-vous cela?

Si le programme de test de l'étudiant utilise des lectures bloquantes, il n'est pas possible pour le système d'exploitation de voir plusieurs requêtes à la fois et d'en optimiser l'ordonnancement. Un bon test devrait soit utiliser plusieurs processus concurrents faisant des lectures ou un processus avec des lectures asynchrones.

- 8.12 Plusieurs versions de Unix utilisent 32 bits pour compter les secondes depuis le début de l'an 1970. Est-ce que cela posera problème? Quand?

Il y a environ  $365.25 * 24 * 60 * 60 = 31557600\text{s}$  par an, soit 136 ans dans 4Gis ( $2^{32}$  secondes). Ceci mène en  $1970 + 136 = 2106$ . Espérons que d'ici là les ordinateurs seront tous à 64 bits, ce qui devrait suffire pour la durée prévue du soleil et même bien au-delà.

- 8.13 Sur le IBM PC original, il fallait écrire dans la mémoire video pendant le court intervalle où le tube cathodique remplaçait son faisceau d'électrons en haut de l'écran et n'était pas en train de lire cette mémoire. L'écran avait 25 lignes de texte de 8 pixels de haut chacune. Le dessin de chaque ligne prenait 63.6us. L'écran était retracé 60 fois par seconde. Pendant quelle proportion du temps est-ce qu'il était possible de mettre à jour la mémoire vidéo?

Le temps requis pour tracer  $25 * 8 = 200$  lignes à 60Hz est de  $200 * 60 * 63.6\text{us} = 763\text{ms}$  à chaque seconde, ce qui laisse 23.7% du temps disponible.

- 8.14 Les concepteurs d'un système ont estimé que la souris peut se déplacer au maximum à 20cm/s. Si la résolution (déplacement minimal rapporté) de la souris est de .1mm et chaque déplacement est codé sur 3 octets, quel est le débit maximal que peut générer la souris?

Si la souris envoie 3 octets à chaque .1mm pour 20cm en une seconde, cela donne  $200\text{mm/s} / .1\text{mm} * 3 \text{ octets} = 6000 \text{ octets/s}$ , ce qui ne présente pas une charge importante.

8.15 La consommation de puissance est une contrainte importante sur les ordinateurs portatifs. Ils essaient de sauver sur leur consommation dès que possible en éteignant certains dispositifs comme le disque dès qu'ils sont inutilisés un certain temps. Un utilisateur note que la consommation est sensiblement réduite lorsqu'il utilise son ordinateur en mode console (pas d'environnement graphique, seulement l'affichage de texte comme sur un terminal) plutôt qu'avec X windows et GNOME ou KDE. Comment expliquez-vous cela?

L'affichage graphique demande beaucoup plus de calcul du CPU ainsi que du GPU. De plus, les logiciels graphiques consomment beaucoup plus de mémoire, ce qui demande des lectures supplémentaires du disque, principalement lors du démarrage de ces grosses applications.