

CCF - ASR7 Programmation Concurrente

Examen final

Numéro d'anonymat : _ _ _ _ _

Yves Caniou, Amaury Maille, Matthieu Moy, Frédéric Suter

Automne 2019

Afin d'obtenir tous les points il vous est demandé de justifier vos résultats. Le barème proposé est susceptible d'être modifié lors de la correction. Il n'est présent que pour vous donner une idée du poids relatif des différentes questions, et/ou la pertinence de la réponse attendue. **La précision et le soin** apportés à vos réponses seront également évalués...

Vous répondrez directement sur le sujet pour les parties Système et Ordonnancement, tout la partie Programmation Concurrente sera faite sur feuille.

I Administration Système (6pt)

I.1 Gestion des utilisateurs et droits (3pt)

```
1  export grp=hackerz
2  groups
3  su -
4  exit
5  su
6  addgroup $grp
7  for i in lala titi roro; do adduser $i ; usermod -aG $grp ; done
8  mkdir -p /home/$grp/view/process
9  echo "touch ../myfic" > /home/$grp/view/process/myscript.sh
10 chmod g+w /home/$grp/view
11 chown -R lala:$grp /home/$grp
12 su lala
13 chmod o-rx /home/$grp/view/process
14 chmod u+s /home/$grp/view/process/myscript.sh
15 sudo roro
16 file /home/$grp/view/process/myscript.sh
17 ls -l /home/$grp/view/process/myscript.sh
18 exit
19 delgroup $grp
20 for i in lala titi roro; do deluser $i ; done
21 rm -r /home/$grp
```

Une liste de commandes tapées au-fur-et-à-mesure par un étudiant est donnée ci-avant. Nous les avons numérotées pour que ça soit plus pratique.

Les questions suivantes ont pour but de comprendre ce que l'étudiant a voulu faire, et d'analyser s'il y est vraiment parvenu.

Merci de répondre précisément et de façon synthétique, par exemple pour la ligne 2 serait : "Affiche les groupes auxquels appartient l'utilisateur qui tape la commande (bien). Pour cela la commande filtre sur le nom de connexion de l'utilisateur les entrées de `/etc/group`, fichier qui contient chaque groupe et ses utilisateurs (parfait). Par exemple si l'utilisateur s'appelle `toto`, la commande devrait afficher notamment `toto`, et possiblement `video audio` sur un PC (bonus)." (mais les questions suivantes requièrent à priori moins de rédaction...).

Q.I.1) - Que fait la commande ligne 3 ?

Demande le mot de passe de root. Permet de passer root, avec les variables d'environnement initialisée comme pour un login

Q.I.2) - Que fait la commande ligne 6 ?

Elle ajoute le groupe `$grp`, donc `hackerz`. Fonctionne car `$grp` a été transformée en variable d'environnement avec `export`, et qu'il y a un `su` de fait avant, donc préservation de l'environnement. C'est la raison pour laquelle l'étudiant s'est déconnecté et reconnecté root.

Q.I.3) - Que fait la commande ligne 11 ?

Change le propriétaire et le groupe propriétaire du nouveau répertoire de façon récursive. Ne change pas les droits

Q.I.4) - Que fait la commande ligne 14 ?

Ajoute le setuid bit au script myscript.sh. Les droits sont maintenant -rwsr-xr-x pour lala :hackerz

Q.I.5) - Que fait la commande ligne 15 ?

la commande roro n'existe pas. Et il faudrait que l'utilisateur soit dans le /etc/sudoers

Q.I.6) - Qu'affiche la commande ligne 17 ?

-rwsr-xr-x lala hackerz

Q.I.7) - Que fait la commande ligne 20?

Elle supprime les utilisateurs lala, titi et roro

Q.I.8) - Qu'affiche la commande ligne 16 (bonus)?

comme on est root, c'est visible. Donne le type de fichier, c-à-d ASCII text (et pas un script). Bonus

I.2 Gestion du système (3pt)

Voici la définition de deux commandes shell dans le fichier `$HOME/bin/script.sh` d'un utilisateur. Elles peuvent s'appeler du terminal comme on le ferait avec une commande aliasée telle `alias l='ls -l'`.

```
function removeExifInfo {  
    exiftool -All="" .  
}  
  
function fb {  
    local list1='ls *jpg 2>/dev/null '  
    local list2='ls *JPG 2>/dev/null '  
    for i in ${list1} ${list2}; do  
        echo ". Converting $i -> r$i"  
        convert ${i} -resize 800x600 r${i}  
    done  
    echo ". Removing Exif info"  
    removeExifInfo  
}
```

Q.I.9) - Comment rendre simplement ces commandes disponibles? (1pt)

Les commandes sont définies dans `script.sh`, donc pour les avoir à disposition il faut lire le contenu de ce fichier dans le shell. On peut le faire pour le shell en cours avec `source`

`/bin/script.sh` (ou `. /bin/script.sh`, c'est pareil). Si on veut ses commandes pour tous les shells ouverts dans le futur, alors il faut ajouter cette ligne dans un fichier d'initialisation comme `/.bashrc`.

Q.I.10) - On me donne un accès `ssh` sur une machine d'adresse IP `ip_machine` pour l'utilisateur `p123456`. Quelle commande est-ce que j'utilise pour copier le script `$HOME/bin/script.sh` sur la machine distante (dans un répertoire quelconque) ? Comment l'administrateur système de la machine distante peut-il vérifier le lendemain que je me suis connecté ? (2pt)

`rsync bin/script.sh p123456@$ip_machine` (ou `scp` qui fait la même chose que `rsync`, mais avec moins de fonctionnalités).

L'administrateur peut vérifier que je me suis connecté en regardant dans `/var/log/auth.log` ou avec la commande `last`.

Q.I.11) - Un utilisateur rends ces fonctions disponibles sur sa machine personnelle, mais ça ne fonctionne pas. Il obtient une erreur "convert : command not found". Que doit-il faire pour que cela fonctionne ? (4pt)

`command -v convert` (ou `which`, ou `type`, ou `whereis` qui font plus ou moins la même chose) sur la machine `src`, qui nous informe que la commande se trouve dans `/usr/bin/convert`. `dpkg -S /usr/bin/convert` nous permet de connaître le paquet qui contient ce fichier, qui est `imagemagick`. Faire un `apt update` sur sa machine. Faire un `apt install imagemagick`.

On obtiendrait un message d'erreur similaire pour un exécutable existant sur la machine

mais pas dans le `$PATH`, mais ce n'est pas le cas ici car `convert` est une commande installée dans un des répertoires standard pour les exécutables par le gestionnaire de paquets.

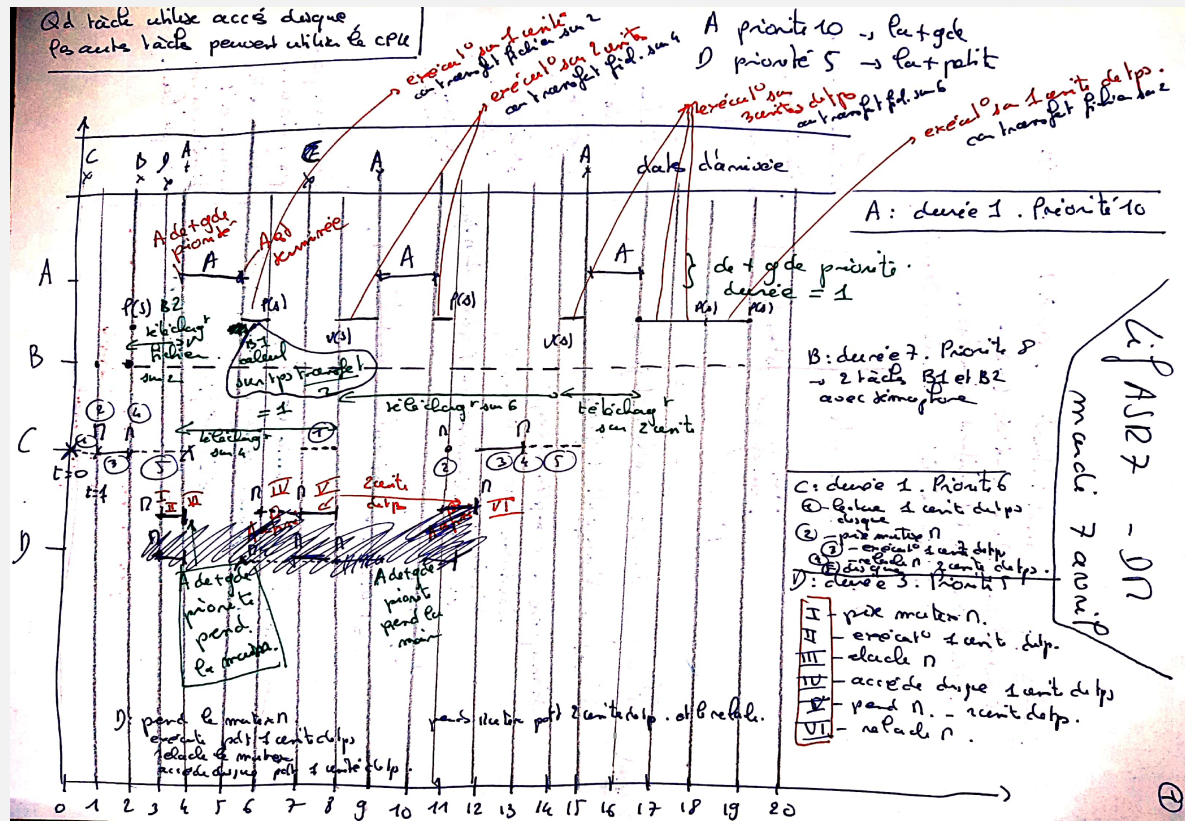
II Ordonnancement avec priorités (4pt)

Nous utilisons un ordonnancement préemptif avec priorité¹ qui se tient à chaque unité de temps sur un système monoprocesseur. Nous allons utiliser un jeu de tâches dans un système préemptible.

Tâche	Date(s) d'arrivée(s)	Priorité	Durée	Remarque
A	4, 9, 15	10	1	Ponctuelle
B	2	8	7	Ponctuelle. 2 tâches B1 et B2 partagent un sémaphore initialisé à 0 et sont lancées telles que : B1 exécute en boucle { P() et calcule pendant un peu moins du temps de transfert du fichier divisé par 2 } ; B2 exécute en boucle le { téléchargement d'un fichier puis V() } : le premier fichier demande un peu moins de 2 temps de transfert, le deuxième fichier en demande moins de 4, le troisième moins de 6, le quatrième moins de 2.
C	0, 7	6	1	Ponctuelle. À chaque fois, elle commence par faire une lecture disque pendant moins de 1 unité de temps, puis elle prend le mutex M, doit s'exécuter pendant moins de 1 unité de temps, relâche le mutex M et accède au disque pendant moins de 2 unités de temps.
D	3	5	3	Ponctuelle. Elle commence par prendre le mutex M, doit s'exécuter pendant un peu plus de 1 unité de temps et relâche le mutex M, accède au disque pendant moins de 1 unité de temps et prend le mutex M pendant moins de 2 unités de temps et le relâche.

Q.II.1) - Faire l'ordonnancement de ces tâches sur 20 unités de temps.

1. Plus la valeur de priorité est importante, plus la tâche est prioritaire



Q.II.2) - Quelles sont toutes les observations que vous pouvez faire?

Au temps 2, P(s) possible ou non. Sinon se fait en 5 et le calcul démarre immédiatement.
idle sur 0-1, 2-3 13-14.

temps 1-2, C prend mutex.

temps 3-6 puis 7-12, D possède mutex.

temps 11, mutex pris par C et endormie car D le possède.

termine à 19.

Q.II.3) - Quel est le temps de réponse (ou latence) de chaque tâche sur l'intervalle demandé

(c'est-à-dire le maximum des durées constatées pour chaque tâche) ?

- A : ...
- B : ...
- C : ...
- D : ...

- A : 1
- B : $19-2=17$
- C : $\max(4-0, 15-7)$
- D : $12-3$

III Programmation concurrente (Vous répondrez sur la copie.)

III.1 Cours (6pt)

Q.III.1) - Rappelez ce qu'est un sémaphore. (2pts)

Cf. le premier CM

Q.III.2) - Peut-on implémenter un sémaphore avec un moniteur de Hoare. Si oui, expliquez (code), si non, argumentez précisément. (4pts)

Cf. le premier CM, et l'exemple du compteur positif avec un moniteur de Hoare.

III.2 Et le travail en collaboration fût (4pt)

Avec 2 amis, c'est séance cuisine chez grand-mère : préparation de Kimchi, du chou chinois pimenté et légèrement fermenté dans de la saumure (eau salée) tant prisé en Corée, avec une recette un peu adaptée ici.

Pour cela, chacun a amené un bocal pour stocker sa préparation, et un bol d'ingrédients nécessaires en quantité suffisante à la préparation : un bol de feuilles de chou préparées, un bol de daïkon et de carottes coupés en julienne, un bol de préparation de gingembre/pomme/-poire/piment/ail/cébettes.

Il faut éviter le gâchis :

- Les 3 amis se mettent d'accord pour exécuter les mêmes actions, entre eux, et de façon répétée en **étapes**.
- À chaque étape, un ami a besoin de chacun des ingrédients : une fois servi, il place le daïkon et la mixture dans la feuille, la plie et la stocke dans son bocal.
- Et personne ne doit se servir en même temps, car il ne faut pas renverser. Ils demandent donc à la grand-mère d'agir de la façon suivante :

- Lorsqu'il n'y a rien sur la table, elle prend au hasard 2 des ingrédients et les place sur la table.
- L'ami qui possède le 3e ingrédient confectionne alors sa feuille avec les 3 ingrédients, et une fois pliée, débarrasse la table en prenant le temps de déposer proprement la feuille pliée dans son bocal.

On va modéliser ceci à l'aide de 4 sémaphores.

Q.III.3) - Dessinez un chronogramme de 4 étapes successives : vous représenterez les 3 amis et la grand-mère, et les actions effectuées par chacun.

Q.III.4) - Donner l'algorithme de la grand-mère (on remarquera que choisir 2 ingrédients est équivalent à choisir un ami donné).

```
On initialise T à 1. On initialise le tableau de sémaphores S[3] à 0.  
while true { P(T) ; Choisit le xe ami V(S[x]) ; }
```

Q.III.5) - Donner l'algorithme de l'ami i .

```
while true { P(S[i]) ; Faire la feuille, la plier. V(T) ; Déposer la feuille proprement dans  
le bocal }
```

Q.III.6) - De quel type de problèmes classiques vu en cours appartient celui-ci ?

C'est une variante du dîner des philosophes

Q.III.7) - Pour que l'après-midi cuisine soit une réussite, il faut que tout le monde puisse préparer une quantité à peu près égale de Kimchi. En conséquence, que doit-on supposer dans les processus décrits avant ?

Il faut simplement avoir une distribution uniforme lors des tirages au sort effectués par l'arbitre/la grand-mère.