

## Examen - Systèmes et Réseaux 1

### Licence 3 Informatique

---

Durée : 2h. Documents personnels autorisés. Le barème est indicatif.

---

#### Exercice 1: Commandes et Système de fichiers (6 pts)

1. Un système de fichiers utilise des blocs de 32ko et des numéros de blocs de 16 bits. Chaque i-nœud contient 7 entrées pour des blocs directs, 1 entrée pour un bloc indirect, 1 entrée pour un bloc doublement indirect et 1 entrée pour un bloc triplement indirect.
  - (a) Quelle est la taille maximale d'un fichier ?
  - (b) Dans ce cas précis, combien de blocs contiennent des données? des adresses de blocs ?
  - (c) Quelle est la taille maximum pour qu'un fichier ne passe pas par une triple indirection ?
2. Le dossier courant contient les répertoires TD1, TD2 ... chacun contient un ou plusieurs fichiers dont le nom commence par CR suivi de la date de rédaction (ex : CR\_12mars2019.pdf).
  - (a) Donner la commande qui permet de créer dans le répertoire courant le fichier `listeCR` qui contienne la liste de tous les compte-rendus.
  - (b) Ajouter à ce fichier le nombre de comptes rendus fait en octobre 2019.
  - (c) Afficher sur le terminal le nombre de fichiers dans TD1 qui peuvent être lus mais pas modifiés par l'utilisateur.

#### Exercice 2: Programmation système (7 pts)

1. Écrire un programme `awk` qui affiche un fichier en inversant l'ordre des lignes (en commençant par la dernière ligne pour finir par la première).
2. Écrire le programme qui fait la même chose en `shell`. On pourra utiliser les commandes `head` et `tail` (par exemple `head -n 10 fic` affiche les 10 premières lignes du fichier `fic` ou bien lues sur l'entrée standard si `fic` n'est pas présent et `tail -n 10 fic` affiche les 10 dernières).
3. Soit le fichier `C exo.c` qui fait appel à la fonction `lire_entier()` qui est définie dans le fichier `lire.c`.
  - (a) Écrire un fichier `Makefile` qui permet d'obtenir l'exécutable `exo` à partir des codes `exo.c` et `lire.c`.
  - (b) On suppose maintenant que la fonction `lire_entier()` fait appel à la fonction `ecrire_entier()` définie dans le fichier `ecrire.c` et que l'on veut construire plusieurs "versions" `exo1`, `exo2`, ... basées sur les fichiers `exo1.c`, `exo2.c`, ... Écrire un fichier `Makefile` générique qui permet d'obtenir l'exécutable `exoi` en lançant la commande `make exoi`.
  - (c) On suppose que l'on a exécuté la commande `make exo1` avec succès une première fois puis que l'on modifie `ecrire.c`. Quelles commandes de compilation seront lancées si l'on exécute de nouveaux `make exo1` ?

Examen Systèmes et Réseaux 1, suite

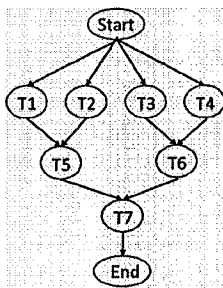
**Exercice 3**

Répondez aux questions suivantes par VRAI ou FAUX. *Vous devez écrire votre réponse dans le tableau ci-dessous et rendre l'énoncé avec votre copie.* Toute réponse fautive vous pénalise car elle annule les points d'une réponse juste.

Question	Votre réponse (Vrai ou Faux)
1 Sous Unix, la commande shell "KILL" sert exclusivement à terminer ou tuer des processus	
2 Une section critique représente un segment de code où le programme peut accéder à une variable ou une ressource partagée.	
3 Un thread est appelé processus léger parce qu'il peut contenir un nombre limité d'instructions	
4 Un sémaphore S peut être utilisé par un processus P1 pour envoyer des données à un autre processus P2.	
5 Dans un modèle Lecteurs – Rédacteurs, plusieurs rédacteurs peuvent écrire en même temps dans un fichier F.	
6 La commande Shell "TRAP" permet à un processus d'envoyer un signal à un autre processus.	

**Exercice 4**

On considère le graphe de tâches ci-dessous :



**Questions**

- En utilisant les concepts de blocs séquentiels BEGIN-END et/ou de blocs parallèles PARBEGIN-PAREND, écrire en pseudo code les tâches (processus) T1, T2, ..., T7.
- En utilisant les sémaphores pour la synchronisation, écrire en pseudo code les tâches T1, T2, T3, T7

**Exercice 5**

On considère trois variables X, Y, Z et un sémaphore S définis comme suit : Int X=1, Y=0, Z=0  
 Sémaphore S ; Sem-Init (S, 2); /\*La valeur initiale du sémaphore S est égale à 2 \*/  
 Soit 3 tâches T1, T2 et T3 définies comme suit :

**Tâche T1**  
 P(S);  
 X = X + 1  
 Y = Y + 1  
 Z = Z + 1  
 V(S)

**Tâche T2**  
 P(S);  
 P(S)  
 X = X + 2  
 Y = Y + 2  
 Z = Z + 2  
 V(S)

**Tâche T3**  
 P(S);  
 P(S)  
 X = X + 3  
 Y = Y + 3  
 Z = Z + 3  
 V(S)  
 V(S):

**Questions**

- Peut-on avoir une situation d'interblocage des tâches ?
- En tenant compte de votre réponse en a) ci-dessus, quelles sont les valeurs possibles des variables X, Y, Z à la fin de l'exécution des 3 tâches ?

*212*