

Durée 2 heures, tous documents issus du cours autorisés (livres interdits)

Sujet recto/verso, le barème est donné à titre indicatif

*Si vous êtes amené à faire des hypothèses, indiquez-les clairement sur la copie*

**Exercice 1 — 10pts :**

1. Soit le fichier IMG\_777.jpg listé ci-dessous :  
ldupont@rubis:~\$ ls -l IMG\_777.jpg  
-rw-rw-r-- 1 ldupont ldupont 1472187 oct. 4 2022 IMG\_777.jpg
  - (a) Quel est le propriétaire du fichier ?
  - (b) Est-ce qu'un utilisateur du même groupe que le propriétaire peut supprimer le fichier IMG\_777.jpg ? Justifier.
  - (c) Quelle commande faut-il exécuter pour donner les droits de modification sur le fichier IMG\_777.jpg aux utilisateurs qui ne sont pas du même groupe que son propriétaire ?
  - (d) Avec des blocs de données de 4Ko et des numéros de blocs codés sur 4 octets, le fichier IMG\_777.jpg peut-il être adressé en deux indirections au maximum ? Justifier.
2. Le répertoire Fouillis contient un grand nombre de fichiers que son propriétaire souhaite ranger ; d'abord en séparant les petits fichiers (taille inférieure à 1 Mo) des grands, puis les rangeant par année et mois de dernière modification. Pour cela, écrire en shell/AWK, les scripts suivants :
  - (a) listepetit qui liste dans le fichier de nom small.lst les noms des fichiers de petite taille ( inférieure à 1Mo) du répertoire courant.
  - (b) separe qui crée les répertoires SMALL et BIG et déplace les petits fichiers dans SMALL et les grands dans BIG.
  - (c) rangeparannee <n> qui déplace tous les fichiers de l'année n du répertoire courant dans un sous-répertoire (créé si besoin) ANNEE\_n.
  - (d) rangeparmois <m> qui déplace tous les fichiers du répertoire courant dans des sous-répertoires (créés si besoin) MOIS\_m.
  - (e) rangetout qui va ranger les fichiers du répertoire courant dans les sous-répertoires SMALL et BIG puis par année et mois.

Suite au verso...

## Exercice 2 — 10pts :

Un travail à réaliser comporte une tâche initiale globale (notée  $TI$ ), deux tâches  $T1$  et  $T2$  chacune décomposée en 3 tâches ( $T1a$ ,  $T1b$ ,  $T1c$  et  $T2a$ ,  $T2b$ ,  $T2c$ ) et une tâche finale  $TF$ . Les tâches  $T1$  et  $T2$  peuvent être réalisées simultanément avec les réserves suivantes, où “début” et “fin” indiquent les instants de début et de fin d’un travail donné :

- début( $T2a$ )  $\geq$  fin( $T1a$ ),
- début( $T1c$ )  $\geq$  fin( $T2b$ ).

1. Dessiner le graphe de dépendence des tâches.
2. En supposant que l’on travaille sur une machine mono-processeur, donner deux exemples d’exécution où les tâches  $T1i$  et  $T2j$  ne sont pas exécutées dans le même ordre.
3. On souhaite implanter ce travail à l’aide de trois processus (un père et deux fils), le père réalisant les tâches  $TI$  et  $TF$  et les fils réalisant les travaux  $T1$  et  $T2$ ; la synchronisation étant effectuée par des signaux.
  - (a) Décrire le protocole à mettre en place (signaux utilisés, de qui vers qui, à quelles occasions, ...). Un schéma sera bienvenu.
  - (b) Donner le code en shell ou C d’un des fils.
4. On souhaite maintenant implanter ce travail à l’aide de deux processus (un père et un fils), le fils réalisant le travail  $T2$  et le père le travail  $T1$ , la synchronisation étant réalisée grâce à des tubes (on utilisera la propriété de lecture bloquante des tubes).
  - (a) Décrire le protocole à mettre en place.
  - (b) Donner, en langage C, le code du processus père.
5. On souhaite finalement implanter ce travail à l’aide de quatre processus (un par grande tâche  $TI$ ,  $T1$ ,  $T2$ ,  $TF$ ), la synchronisation étant réalisée grâce à des sémaphores. Donner le pseudocode des quatre programmes en utilisant les instructions  $P(Semaphore)$ ,  $V(Semaphore)$  (vous pouvez utiliser autant de sémaphores que nécessaire, en n’oubliant pas d’indiquer leur valeur initiale).