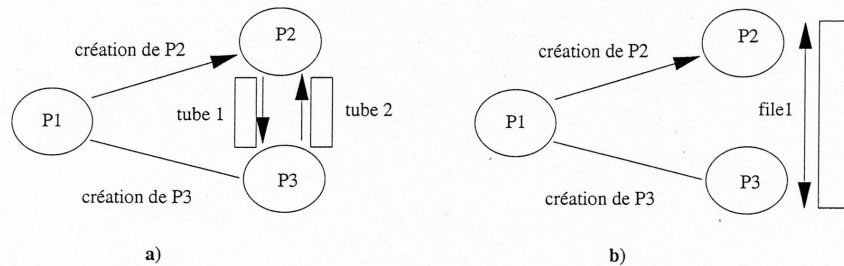


Final LO41 - 25/06/2002

Durée 2 heures ; notes de cours, de TD et de TP autorisées ; 1 copie par partie

PARTIE A (dans la mesure du possible, répondez sur les feuilles de programmes annexes)



1. Utilisation des tubes pour communiquer (5pts)

Écrire un programme C correspondant au schéma a) de la figure précédente et selon les exigences suivantes :

- . Un processus P1 crée 2 processus (P2 et P3) et 2 tubes de communication (tube1 et tube2). Il attend la fin de ses 2 fils, écrit sur la sortie standard le nom de chaque processus qui se termine, puis se termine lui-même.
- . Le processus P2 écrit dans le tube1 son numéro de processus, puis lit dans le tube2 le numéro de processus que lui a envoyé le processus P3.
- . Le processus P3 écrit dans le tube2 son numéro de processus, puis lit dans le tube1 le numéro de processus que lui a envoyé le processus P2.

2. Utilisation des IPC files de messages pour communiquer (4pts)

Écrire un programme C correspondant au schéma b) de la figure précédente et selon les exigences suivantes :

- . Un processus P1 crée 2 processus (P2 et P3) et 1 file de messages (file1).
- . Il attend la fin de ses 2 fils, écrit sur la sortie standard le nom de chaque processus qui se termine, détruit la file de messages, puis se termine lui-même.
- . Les processus P2 et P3, comme dans l'exercice précédent, s'écrivent et lisent mutuellement leurs numéros de processus.

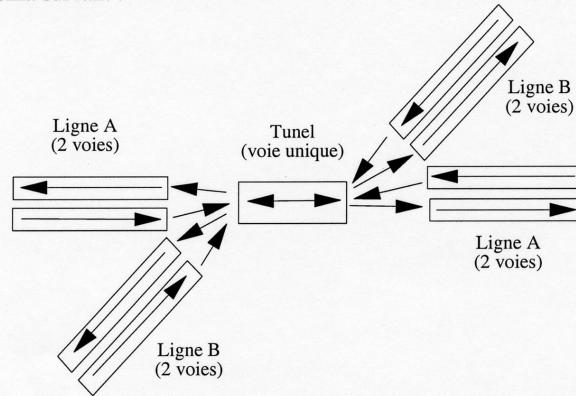
3. Utilisation des signaux (3pts) (continuez à compléter les feuilles de programmes annexes)

Complétez le programme C précédent pour répondre aux exigences suivantes, concernant la destruction de la file de message :

- . En cas de réception de `Ctrl-C` le processus P1, détruit la file de message et se termine.
- . Quoiqu'il arrive (blocage d'un processus ou autre ...), le processus P1 doit pouvoir détruire la file de message au bout de 5 secondes (vous utiliserez la fonction `alarm()`).

PARTIE B

On désire réaliser une simulation de la gestion d'un tunnel à haut risque dans le domaine ferroviaire, selon le schéma suivant :



Pour franchir une montagne, les trains passant sur deux lignes bidirectionnelles distinctes doivent emprunter un tunnel qui ne comporte qu'une seule voie. On se propose de faire la gestion de ce tunnel en évitant bien sûr tout risque de collision.

1. 1ère simulation : mode de gestion supervisée (3pts)

On considère un processus superviseur, 4 processus voies et des processus trains. Les processus trains sont créés aléatoirement pour les besoins de la simulation, et se terminent après avoir franchi le tunnel. Les processus trains font leur demande de franchissement de tunnel à leur processus voie respectif. Le processus superviseur reçoit des demandes de passage des voies. Il répond par un signal d'accord. Après avoir fait une demande de passage, un train se met en attente sur sa voie. Ecrivez sous forme algorithmique le processus superviseur, un processus voie, un processus train.

2. 2ème simulation : mode de gestion distribuée (3pts)

On considère 4 processus voies qui se partagent de l'information et des processus trains sous les mêmes conditions que dans la question 1. Les processus trains font leur demande de franchissement de tunnel à leur processus voie respectif. Le processus voie consulte une variable partagée par tous les processus voies qui indique si un train a eu l'autorisation de passer dans le tunnel. Après avoir fait une demande de passage, un train se met en attente sur sa voie. Ecrivez sous forme algorithmique un processus voie. Vous pouvez utiliser les primitives P et V vues durant l'UV.

3. 3ème simulation : mode de gestion distribuée-contrôlée (2pts)

Par rapport au mode distribué précédent, on désire pouvoir contrôler les passages dans le tunnel pour gérer des situations telles que : des demandes prioritaires, un problème dans le tunnel, etc. Pour cela on rajoute un processus contrôleur qui met à jour les autorisations de passage des 4 voies par l'intermédiaire d'un tableau de 4 cases. Chaque case correspond à l'autorisation de passage d'un train sur une voie donnée.

Ecrivez sous forme algorithmique le processus contrôleur et un processus voie. Vous pouvez utiliser les primitives P et V vues durant l'UV.