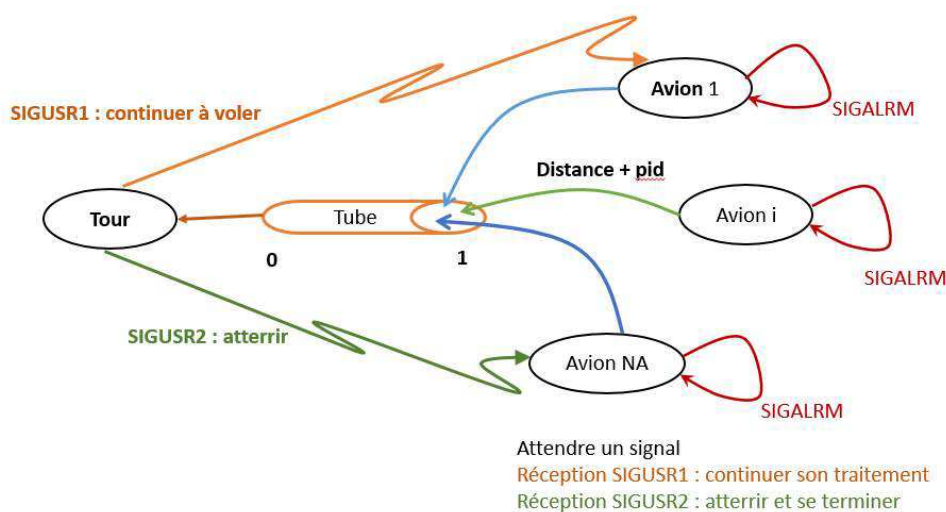


Contrôle terminal – Session 2 – 19 juin 2017

Eléments de solution

Exercice 1 : Processus UNIX (15 pts)

1. Faire une **description schématique** claire, précise et commentée de la solution proposée qui utilisera **signaux** et **tubes**. (4 pts)



2. Écrire le **code** d'un processus **avion** et du **processus** **tour** ainsi que le code de la fonction **main** qui initialise l'application comportant la tour et NA avions. (11 pts)

Main : 3 pts

Créer tube
Protéger fils contre signaux
Créer NA fils avions (NA paramètre du main)
« Lancer » processus tour

Avion : 4 pts

Orienter tube (fermer lecture)
Gérer SIGALRM (se protéger en associant fonction de traitement, armer délai)
Gérer SIGUSR1 pour continuer à voler (fonction qui ne fait rien)
Gérer SIGUSR2 pour atterrir (fonction qui affiche message et positionne l'indicateur continuer à faux pour terminer boucle de traitement) :
Boucler tant qu'il faut continuer à voler
Écrire (pid + distance) dans le tube
Attendre un des 2 signaux en provenance de la tour

Tour : 4 pts

Orienter tube (fermer écriture)
Boucler tant qu'il reste un avion en vol
Boucler jusqu'à lire autant de messages que d'avions en vol
Déterminer qui est le pid de l'avion le plus proche
Envoyer le signal SIGUSR2 à l'avion qui doit atterrir et le signal SIGUSR1 à tous les autres
Décrémenter le nombre d'avions en vol

Exercice 2 : Synchronisation par sémaphores (5 pts)

1. Donnez les affichages possibles après le lancement parallèle de ces processus. (2 pts)

ABCD	ABDC	ACDB	ACBD	ADCB	ADBC
CABD	CADB	CDAB			
DABC	DACB	DCAB			

2. En utilisant un ou plusieurs sémaphores, **synchronisez** ces processus de manière à **restreindre** les affichages possibles, à chaque cycle, à ACDB **ou** ADCB. (3 pts)

init(semB, 0) Processus P1 { Boucler { Afficher 'A' ; V(semC) V(semD) P(semB) P(semB) Afficher 'B' ; } }	init(semC, 0) Processus P2 { Boucler { P(semC) Afficher 'C' ; V(semB) } }	init(semD, 0) Processus P3 { Boucler { P(semD) Afficher 'D' ; V(semB) } }
---	---	---