

## Présentation

Le but de ce TP est la mise en application des concepts liés à la synchronisation et la communication des processus à travers l'utilisation d'outils offerts par le système d'exploitation UNIX. On se propose d'implémenter un système imaginé d'un problème de la vie quotidienne comme suit :

Il s'agit de la gestion d'affectation d'organes humains à des malades dans un hôpital spécialisé pour des fins d'implantations. Imaginons un *hôpital* et un ensemble de deux populations, les *malades* et les *donneurs* d'organes. Les malades sont donc des patients qui souhaitent qu'on leur plante un organe humain suite à la défaillance de leur ou de son ablation à cause par exemple d'accident. Ces malades sont divisés en deux classes : ceux qui ne nécessitent pas d'hospitalisation, en l'occurrence ceux qui demandent par exemple un bras ou une jambe ; l'autre classe est constituée de malades qui sont hospitalisés, en l'occurrence ceux qui demandent un organe vital par exemple un cœur, un poumon ou un rein. La durée de vie des malades de cette dernière classe est limitée dans le temps s'ils ne reçoivent pas l'organe au moment voulu. Les donneurs sont des personnes qui viennent de mourir normalement ou suite à un accident par exemple ; en général les organes qui proviennent de personnes mortes après un accident sont conseillés par rapport à d'autres étant donné qu'il s'agit de personnes qui étaient souvent en bonne santé.

## Modélisation

Les éléments qui interviennent dans le problème sont :

- \* Les organes : Ils sont de deux types, organes vitaux (au nombre de  $N1$ , par exemple cœur, poumon ...) et les organes non vitaux (au nombre  $N2$ , par exemple bras, jambe...).

- \* Les donneurs : Ils sont modélisés par un seul processus cyclique *donneur* qui génère de temps en temps un organe (choisi de manière aléatoire,) caractérisé par (*type, nom*) où *type* est : (*vital, non vital*) et *nom* : (*poumon, bras, ...*) qu'il envoie symboliquement dans un tampon *Torganes* de taille  $N$ . Un générateur aléatoire peut être utilisé pour déterminer la durée d'attente avant de fournir un prochain organe.

- \* Les malades normalement au nombre indéterminé : Ils sont représentés uniquement par deux processus. Une partie du travail de chacun des deux processus est de générer à chaque fois une requête représentant la commande d'un malade et de garder une copie dans une file locale. Cette file sert lors de la réponse d'implantation d'un organe pour déterminer la requête concernée. Un générateur aléatoire peut être utilisé pour déterminer la durée d'attente avant de générer une prochaine requête. Le reste du comportement logique de chaque processus est :

- Processus *malade\_critique* : L'enregistrement de chaque requête à l'hôpital se fait "en allant" à l'hôpital. La requête comprend : (*type de l'organe correspondant à sa catégorie, nom de l'organe, nom du malade (i.e. un numéro de séquence)*). Le patient reste à l'hôpital jusqu'à ce qu'on lui plante l'organe correspondant. Pratiquement, l'enregistrement se fait dans une file de messages *QCritique* FIFO.

- Processus *malade\_non\_critique* : L'enregistrement de chaque requête à l'hôpital se fait "en allant" à l'hôpital. La requête comprend : (*type de l'organe correspondant à sa catégorie, nom de l'organe, nom du malade (i.e. un numéro de séquence)*). Le patient revient chez soi jusqu'à ce qu'on lui plante l'organe correspondant. Pratiquement, l'enregistrement se fait dans une file de messages *QNonCritique* FIFO.

- Le *chirurgien* : Il reste à l'écoute des malades et des donneurs pour réaliser le service d'implantation des organes aux malades. Il est représenté par un processus cyclique *chirurgien* dont le rôle est d'attendre les requêtes (file *QCritique* et *QNonCritique*) respectivement des processus malades (*malade\_critique* et *malade\_non\_critique*), de commander les organes demandés (en envoyant un message à travers un tube au processus donneur correspondant) et à chaque réception d'un organe, réaliser l'implantation de celui-ci. L'implantation d'un organe obtenu au malade concerné est représentée par l'envoi d'une réponse contenant (*type organe, nom organe, nom malade*) dans une file de messages *QImplantation* accessible par chacun des deux processus malades. Ces derniers scrutent régulièrement cette file pour récupérer éventuellement une réponse appropriée pour noter sa satisfaction.

## Travail demandé

Il s'agit de synchroniser tous ces processus pour réaliser la fonction décrite ci-dessus pour chacune des deux étapes indiquées ci-dessus. Le système doit comprendre une interface (textuelle seulement, sous forme d'impressions) permettant de suivre et de comprendre le fonctionnement des différentes entités du service objet de ce TP.

Les processus du système sont définis pour fonctionner indéfiniment ; cependant pour des fins de tests, on suppose que les deux processus *malade\_critique* et *malade\_non\_critique* génèrent seulement et respectivement  $M1$  et  $M2$  requêtes. En conséquence, le processus *donneur* fournit  $(M1+M2)$  organes puis conclue et le processus *chirurgien* traite  $(M1+M2)$  requêtent puis conclue.

## Connaissances requises sous Unix

- Partage de mémoires
- Synchronisation par sémaphores
- Communication par tubes.
- Communication par messages (files de messages)

## Echéance

04 Décembre 2025