

Rapport de TP

Architecture des systèmes d'exploitation

Prénom NOM

Novembre 2021

Ce document vous suggère une trame de rapport de TP.

Vous pouvez utiliser le source \LaTeX de ce document comme base (voir également la cible `rapport.pdf` dans le fichier `Makefile`), mais vous pouvez utiliser tout autre outil de votre choix, à condition de rendre un rapport en PDF.

1 Introduction

Rappelez brièvement le sujet et esquissez les principes de votre solution (une dizaine de lignes maximum).

2 Structure de données

2.1 Structures de données partagées

Décrivez ici la structure de votre segment de mémoire partagée, ainsi que toutes les structures annexes que vous y utilisez. Pour chaque donnée, indiquez son nom, sa nature (entier, sémaphore, tableau de X...), sa valeur initiale, et décrivez en quelques mots son usage.

Par exemple, si le vaccinodrome disposait d'un distributeur de gel hydroalcoolique à l'usage des patients, on trouverait la description suivante :

Champ	Type	Init	Description
<code>dg</code>	<code>distrib_t</code>	–	distributeur (cf. documentation de <code>libhygiene</code>)
<code>sm_dg</code>	<code>asem_t</code>	1	sémaphore utilisé pour synchroniser l'accès au distributeur

On considère que tout ce qui est décrit dans cette section existe depuis la création du vaccinodrome jusqu'à sa fermeture.

2.2 Structures de données non partagées

Indiquez toutes les données conservées par les processus mais non partagées. La forme est la même que dans la section précédente. Précisez aussi à quel moment ces données sont créées et initialisées.

3 Synchronisations

Chacune des sous-sections suivantes doit décrire un aspect de la synchronisation. Elles ont toutes la même organisation, et doivent contenir deux ou trois parties clairement délimitées.

1. La première partie décrit tous les objets impliqués dans la synchronisation de l'accueil des patients dans le vaccinodrome. Il s'agit des données « normales » (par exemple un entier) et des sémaphores. Pour chacun de ces éléments, précisez si il est rattaché au vaccinodrome globalement ou à un patient, ou encore à un médecin. Par exemple, si le vaccinodrome était muni d'un distributeur de gel hydroalcoolique grâce auxquels les patients peuvent se nettoyer les mains, il faudrait mentionner :

<code>sm_dg</code>	<code>asem_t</code>	le sémaphore représentant le nombre de distributeurs <u>du vaccinodrome</u> (voir section 2.1)
<code>monid</code>	<code>int</code>	le nom <u>du patient</u>

2. La seconde partie doit contenir un pseudo-code de ce qu'exécute chaque acteur impliqué (le patient, le médecin, autres). Seuls les aspects importants doivent être présentés : opérations sur les sémaphores, affectation de valeurs aux variables partagées... Le reste (par exemple les messages liés à

`DEBUG_OUTPUT`) ne doivent pas apparaître ici. Dans l'exemple des distributeurs de gel, le pseudo-code serait :

```
// Ce code est exécuté par le patient
P (sm_dg)
se_laver_les_mains (monid)
V (sm_dg)
```

3. La troisième partie contient toutes les remarques que vous jugez pertinentes. Vous pouvez par exemple y préciser les conditions de concurrence, ou les éventuelles limitations ou propriétés remarquables de votre solution.

Dans l'exemple des distributeurs de gel, vous pourriez préciser que vous avez pensé à implémenter à la place une attente limitée dans le temps pour les patients qui portaient des gants en entrant (en expliquant comment vous auriez fait), mais que cela aurait compliqué le code pour un intérêt limité.

Vous pouvez ajouter des sous-sections si vous avez d'autres synchronisations importantes, ou fragmenter celles qui vous sont proposées ci-dessous.

3.1 Arrivée d'un patient

Expliquez dans cette sous-section, conformément aux indications ci-dessus, tout ce qui se passe lorsqu'un patient arrive, avant qu'il interagisse avec un médecin (si le vaccinodrome est ouvert) ou reparte. Mentionnez toutes les informations partagées qui sont utilisées et toutes les synchronisations nécessaires.

3.2 Arrivée d'un médecin

Expliquez dans cette sous-section tout ce qui se passe lorsqu'un médecin arrive, avant qu'il interagisse avec un patient.

3.3 Interactions entre patients et médecins

Expliquez ici comment patients et médecins sont « mis en contact ». Expliquez la stratégie que vous avez choisie, c'est-à-dire comment 1) soit un patient trouve un médecin dès qu'un médecin est disponible, 2) soit un médecin trouve un patient dans la salle d'attente. Expliquez également ce qui se passe entre le moment où un médecin et un patient sont mis en contact, et le moment où le patient vacciné peut quitter le vaccinodrome et le médecin s'occuper d'un (éventuel) autre patient.

3.4 Fermeture du vaccinodrome

Expliquez les détails de ce que fait le programme `fermer` (et seulement ce programme), y compris lorsqu'il reste des patients et/ou des médecins.

3.5 Patients après fermeture

Expliquez ce qui se passe pour un patient assis dans la salle d'attente après la fermeture. (Notez que le cas du patient qui arrive après la fermeture doit avoir été décrit dans la section 3.1.)

3.6 Médecins après fermeture

Expliquez ce qui se passe pour un médecin après la fermeture. (Le cas du médecin qui arrive après la fermeture doit avoir été décrit dans la section 3.1.)

4 Remarques sur l'implémentation

Placez ici toute remarque relative à vos choix d'implémentation (si vous en avez). Par exemple : si vous avez placé des temporisations particulières pour vérifier que certains cas de concurrence étaient gérés correctement, expliquez cela dans cette section. Toute autre remarque qui aide à comprendre votre code est à placer ici.

5 Conclusion

Tirez les conclusions de votre projet : limites de votre implémentation, difficultés particulières, subtilités dont vous êtes fiers, etc.