

Système d'exploitation

Projet : Simulation d'épidémie

I – Objectifs

Les objectifs du projet de simulation d'épidémie étaient de réaliser un simulateur avec les moyens proposés par le système Unix (threads, processus, signaux...).

II – Structure du simulateur

Le simulateur est composé de 4 programmes :

- epidemic_sim : il s'agit du programme principal, il gère les échanges avec le timer, la press_agency et le citizen_manager. Il s'occupe aussi de mettre à jour l'interface graphique et de stocker les données de la simulation dans evolution.txt à l'aide d'un processus fils
- citizen_manager : il contrôle et synchronise les threads des différents citoyens
- press_agency : affiche les dépêches envoyées par les journalistes
- timer : permet à epidemic_sim de savoir quand débiter un nouveau cycle de simulation

III – Fonctionnement de la simulation

Via des tubes epidemic_sim peut envoyer les instructions, INIT, UPDATE et DESTROY aux autres programmes.

- Avant l'initialisation
 1. epidemic_sim crée les tubes et les ouvre
 2. citizen_manager, press_agency et timer ouvre les tubes
- Initialisation

1. epidemic_sim s'initialise : il crée la carte dans la mémoire partagée
 2. epidemic_sim envoie INIT à timer
 3. timer s'initialise
 4. epidemic_sim envoie INIT à press_agency
 5. press_agency s'initialise, elle crée les files de messages pour recevoir les dépêches des journalistes
 6. epidemic_sim envoie INIT à citizen_manager
 7. citizen_manager configure l'accès à la mémoire partagée pour pouvoir lire la carte puis crée les threads pour les citoyens et ouvre les files de message pour les journalistes
- Actualisation
 1. epidemic_sim envoie UPDATE à timer
 2. timer renvoie True à epidemic_sim dès qu'il reçoit un signal ALARM
 3. epidemic_sim envoie UPDATE à press_agency
 4. press_agency affiche les dépêches des journalistes
 5. epidemic_sim envoie UPDATE à citizen_manager
 6. les citoyens sont mis à jour dans cet ordre (chaque étape est exécutée séparément par chaque thread puis les threads se synchronisent entre chaque étape) : déplacement (ou non) du citoyen, échange de contamination avec la tuile, contamination des autres citoyens et action spécial (soigner pour un docteur...)
 - Destruction
 1. epidemic_sim envoie DESTROY à timer, citizen_manager et press_agency
 2. timer ferme les tubes et s'arrête
 3. citizen_manager détruit les threads et ferme les tubes
 4. press_agency détruit la file de message et ferme les tubes
 5. epidemic_sim détruit les tubes et sauvegarde le graphique obtenu dans graph.png par l'intermédiaire de son processus fils

IV – Exemple de simulation

