

Université d'État d'Haïti

Faculté Des Sciences

Génie électronique – EN3

Système exploitation

Projet: Simulation de certaines opérations, librement choisies, d'un système d'exploitation

Objectifs: Simuler les opérations sur une représentation conceptuelle du matériel

Pascal JEANTY

Cebien AUGUSTIN

Joseph Jeff FORESTAL

Septembre 2023

Contenu du projet

A- Dossier de conception du système

- 1- Spécifications du système
- 2- Flux opérationnel d'exécution du programme de simulation
- 3- Description des différents cas d'utilisation du système
- 4- Modélisation des principales composantes, pour la simulation, telles que:
 - Mémoire vive (RAM)
 - Processus
 - Mémoire secondaire
 - Système de fichiers
- 5- Les différentes composantes du système et leurs interactions
- 6- Description du générateur d'évènement qui simulera toutes les interruptions que le système d'exploitation aura à traiter tant du point de vue d'évènement provoqué par le matériel tel que le quota que du point de vue logiciel, telle qu'opération d'entrée/sortie par un processus utilisateur.

B- Simulation en Java su système

Créer au moins deux threads qui roulent en parallèle :

- 1- Le générateur d'évènement qui, à partir de la fonction Random() de Java, génère toute une série de chiffres aléatoires qui, arrangés de façon judicieuse simulent les différents appels (system call) au système d'exploitation.
- 2- Le système d'exploitation qui traite toutes les interruptions produites par le générateur d'évènements.

Le programme doit disposer d'un log lui permettant de :

- 1- Mémoriser les différents évènements générés et traités
- 2- Afficher les différents évènements générés et traités

A- Dossier de conception du système

1- Spécifications du système

Dans le cadre de cette simulation d'un système d'exploitation, on choisit de simuler les opérations suivantes d'un système d'exploitation :

- a) La gestions des processus
- b) La gestion de la mémoire
- c) La gestion des interruptions

Le système devra pouvoir:

- Générer, de manière aléatoire, une série de chiffres simulant les différents appels au système d'exploitation.
- Afficher les différents événements générés et traités sur la console.
- Mémoriser les différents événements générés et traités dans un fichier de journalisation (log).

2- Flux opérationnel d'exécution du programme de simulation

Le programme, une fois lancé, au moyen du générateur d'événements:

a) Initialisation :

- Le système d'exploitation démarre en initialisant les structures de données, les composantes du système d'exploitation et les ressources nécessaires pour l'exécution.

b) Exécution du générateur d'évènements :

- Le générateur d'évènements commence à générer aléatoirement des appels système et des interruptions.
- Pour chaque évènement généré, le générateur d'évènements les ajoute à la file d'attente (pris en charge par implémentation des threads).

c) Traitement des évènements par le système d'exploitation :

- Le système d'exploitation surveille en permanence la file d'attente d'évènements pour détecter les nouveaux évènements.
- Lorsqu'un nouvel évènement est détecté, le système d'exploitation le traite en fonction de son type (appel système ou interruption).

d) Changement de contexte (Context switch):

- Lorsque le système d'exploitation traite un évènement, il peut décider de changer de thread (processus) actif (si l'évènement correspond à une interruption matérielle ou un changement d'état du processus).
- Le système d'exploitation effectue le changement de contexte en sauvegardant l'état du processus (thread) actif et en chargeant l'état du nouveau processus à exécuter.

e) Traitement des appels système :

- Lorsqu'un appel système est détecté, le système d'exploitation effectue les opérations demandées par le processus (thread) correspondant.
- Par exemple, dans le cas d'une lecture ou d'une écriture, le système d'exploitation gère l'accès aux données et les transmet au processus demandeur.

f) Gestion de la mémoire :

- Le système d'exploitation alloue et gère la mémoire vive pour les différents processus (threads).
- Si un processus (thread) demande plus de mémoire, le système d'exploitation peut effectuer des opérations de chargement ou de déchargement entre la mémoire vive et la mémoire secondaire.

g) Traitement des interruptions matérielles :

- Lorsqu'une interruption matérielle est détectée (par exemple, fin de quota de temps d'exécution), le système d'exploitation interrompt le processus en cours d'exécution et passe à un autre processus (thread).

h) Enregistrement des évènements dans le journal :

- A chaque étape importante de traitement des évènements, le système d'exploitation enregistre les détails pertinents dans le journal (log) pour suivre les évènements générés et traités.

i) Terminaison du programme :

- Le programme se termine lorsque le générateur d'évènements a généré suffisamment d'évènements ou lorsque le système d'exploitation atteint un état de terminaison spécifié.

3- Les cas d'utilisation du système

a) Démarrage de de la simulation

b) Création d'un nouveau thread (processus) :

- L'utilisateur lance le programme de simulation.
- Le système crée un nouveau thread (processus) pour exécuter l'application et alloue des ressources appropriées (mémoire, temps CPU, registre, fichiers, etc).

c) Gérer les interruptions :

- Le système d'exploitation contenant le programme principal (le thread principal) traite les interruptions générées par le générateur d'évènements.

d) Changer de thread (processus) actif:

- Le système d'exploitation effectue le changement de contexte entre les différents threads (context switch).
- Lorsqu'un thread est interrompu ou se termine, le système d'exploitation sélectionne un nouveau thread à exécuter.

e) Effectuer des opérations d'entrée/sortie :

- Les thread (processus) peuvent demander des opérations d'entrée/sortie (écriture dans le fichier de journalisation(log)).
- Le système d'exploitation gère ces demandes et assure la coordination appropriée pour que l'opération d'écriture dans le fichier de journalisation soit effectuée de manière ordonnée et sans conflits.

f) Gérer les erreurs et les exceptions :

- Le système d'exploitation gère les erreurs et les exceptions qui peuvent survenir pendant l'exécution des threads.
- En cas d'erreur ou d'exception, le système d'exploitation peut tenter de récupérer ou de signaler l'erreur à l'utilisateur.

g) Arrêt (fin) de la simulation :

- La simulation s'arrête lorsque les threads auront fini de s'exécuter et que les évènements ont été enregistrés dans le fichier de journalisation (log).

4- Modélisation des principales composantes pour la simulation

a) La mémoire vive

- On peut modéliser la mémoire vive du système comme un espace mémoire virtuel où les processus (threads) peuvent être chargés et exécutés.
- Pour simplifier on peut utiliser une liste ou un tableau pour représenter la mémoire vive, où chaque entrée correspond à une zone mémoire.
- Chaque processus (thread) sera chargé dans une zone mémoire et pourra accéder à ses données et instructions pendant son exécution.

b) Processus :

- Un processus est une tâche en exécution dans le système d'exploitation. Chaque processus a son propre espace d'adressage et son état d'exécution.
- Pour la simulation, nous utiliserons les threads de java.

c) Mémoire secondaire :

- Pour représenter la mémoire secondaire (disque ou autre mémoire de stockage), nous pouvons utiliser des fichiers sur le système de fichiers de la machine hôte.
- Lorsque les processus (thread) ne sont pas en exécution, leur état peut être sauvegardé dans des fichiers sur le disque dur et rechargé en mémoire vive lorsque le processus doit être rechargé à nouveau.

d) Système de fichiers :

- Pour représenter le système de fichiers, nous pouvons utiliser des structures de données pour stocker les informations sur les fichiers tels que leur nom, leur emplacement, leur taille, etc.
- Nous pouvons également implémenter des opérations de base du système de fichiers telles que la création des fichiers, la lecture, l'écriture, comme c'en est le cas pour le fichier de journalisation (log).

5- Les différents composants du système et leurs interactions

a) Générateur d'événements :

- Cette composante est responsable de la génération aléatoire des appels système simulés.
- Elle génère des appels système de manière aléatoire, tels que lecture et écriture.
- Interactions : le générateur d'événements informe le système d'exploitation des appels système générés en utilisant une file d'attente simulée par la synchronisation des threads en java.

b) Système d'exploitation (OS) :

- Cette composante gère l'exécution des processus (threads) et traite les appels système générés par le générateur d'événements.
- Il s'occupe du changement de contexte entre les processus (threads), alloue la mémoire vive aux processus (threads) et effectue les actions spécifiques en fonction du type d'appel système (lecture, écriture, etc.).
- Interactions : Le système d'exploitation récupère les appels système de la file d'attente d'événements et les traite en fonction du type d'appel.

c) File d'attente des événements (Event Queue) :

- On utilisera la synchronisation des threads java en lieu d'une structure de données (file d'attente ou une liste) pour simuler la transmission des appels système générés par le générateur d'événements au système d'exploitation.
- Le générateur d'événements ajoute les appels système à cette file d'attente, et le système d'exploitation la surveille en permanence pour détecter les nouveaux événements à traiter.

d) Processus :

- Un processus est une unité d'exécution du système d'exploitation, dans notre simulation on simulera un processus par un thread qui lui-même est une entité au sein d'un processus qui peut être ordonnancé pour être exécuté.
- Interactions : Le système d'exploitation gère les processus (threads) en leur allouant de la mémoire vive, en planifiant leur exécution, en passant d'un état à l'autre, etc.

e) Mémoire vive (RAM) :

- La mémoire vive est l'espace mémoire où les processus (threads) sont chargés et s'exécutent.
- Interactions : Le système d'exploitation alloue de la mémoire vive aux processus (threads) pour qu'ils puissent s'exécuter et y stocker leurs données et instructions.

f) Mémoire secondaire (disque dur) :

- C'est la mémoire de stockage permanente du système, utilisée pour stocker les processus (threads) et les données lorsque ceux-ci ne sont pas en cours d'exécution en mémoire vive.
- Interactions : Le système d'exploitation peut sauvegarder l'état des processus inactifs sur le disque dur et les recharger en mémoire vive lorsque nécessaire.

g) Système de fichiers :

- Cette composante gère l'organisation et l'accès aux fichiers du système.
- Elle permet la création, la lecture, l'écriture et la suppression de fichiers par les processus.
- Interactions : Le système d'exploitation utilise le système de fichiers pour gérer les opérations de lecture et d'écriture des fichiers demandés par les processus.

6- Description du générateur d'évènement qui simulera toutes les interruptions que le système d'exploitation aura à traiter tant du point de vue d'évènement provoqué par le matériel tel que le quota que du point de vue logiciel, telle qu'opération d'entrée/sortie par un processus utilisateur.

a) Génération d'événements aléatoires :

- Le générateur d'événements utilise une source de nombres aléatoires, au moyen de la fonction Random() de java, pour créer des événements de manière aléatoire. Cela permet de simuler des situations réalistes où les événements surviennent de manière imprévisible.

b) Types d'événements :

- Le générateur d'événements peut être configuré pour générer différents types d'événements, tels que des interruptions matérielles, des demandes d'opérations d'entrée/sortie par des processus utilisateur, des changements de contexte des threads, etc. chaque type d'événement est associé à une probabilité de génération.

c) Intervalle de génération :

- Le générateur d'événements peut être configuré avec un intervalle de génération (en millisecondes), qui détermine à quelle fréquence de nouveaux événements sont générés.

d) Attributs d'événements :

- Chaque événement généré est associé à des attributs spécifiques en fonction de son type.

e) File d'attente des événements

- Les événements générés sont placés dans une file d'attente. Cette file d'attente est ensuite traitée par le système d'exploitation.

f) Interactions avec le système d'exploitation :

- Une fois qu'un événement aléatoire est généré, il est envoyé au système d'exploitation pour traitement.