

Fiche SE

Etapes nécessaires à la génération d'un programme exécutable à partir d'un programme source écrit en langage C. Vous préciserez en particulier ce que sont un fichier d'entête et une bibliothèque.

Compilation : contrôles lexicaux et syntaxiques, traduction en langage machine des instructions C. Génération d'un programme objet. Ne contient que la traduction du code présent dans le fichier source.

Édition des liens : regroupement en un seul fichier exécutable du code objet des différentes fonctions composant le programme. Ce code peut être contenu dans des fichiers objet indépendants ou dans une bibliothèque (fonctions f...).

Qu'est-ce qu'un processus ?

Image mémoire d'un programme en cours d'exécution. Composé du code, des différentes zones de données (pile, données statiques) et des informations système.

Qu'est-ce qu'un thread ?

Tâche = flot d'instruction pouvant être parallélisés. Unité d'ordonnancement.

RAM ? ROM ?

RAM : mémoire vive, contient des programmes du SE, perd ses données éteint.

ROM : mémoire morte, contient les programmes de test de la carte mère et le BIOS, ne perd pas ses données éteint.

Pourquoi un programme exécutable est-il spécifique à un système d'exploitation et à une architecture matérielle ?

Système d'exploitation : présence de directives spécifiques, en particulier pour le chargement en mémoire, présence d'appels systèmes.

Changer : bibliothèques différentes, format de l'exécutable différent -> refaire au moins la compilation et l'édition des liens

Architecture matérielle (processeur) : le jeu d'instruction de chaque processeur est différent.

Changer : jeu d'instructions différent -> compilation + édition de liens

Qu'est-ce qu'une interruption ? Quel est l'intérêt pour un système multitâches de gérer ses entrées-sorties par interruption ?

Signal envoyé de manière asynchrone au processeur par un dispositif physique (contrôleur d'interruption...) pour l'avertir d'un événement externe (périphérique) ou interne (division par 0, erreur d'adressage, ...)

Un ordinateur dédié à une application unique n'a pas obligatoirement besoin d'un système d'exploitation.

- Quelle est la condition pour que l'ordinateur puisse démarrer dans ce cas ?

- Que devra assurer l'application qui aurait été fait pas le système s'il y en avait eu un ?

Il doit y avoir un programme bootloader dans une mémoire non volatile (comme une ROM ou un disque). Ce programme permet d'initialiser le matériel et de charger une application directement en mémoire pour l'exécuter.

Ce que l'application doit assurer sans système d'exploitation :

L'application devra gérer elle-même toutes les fonctions normalement assurées par un OS, telles que :

La gestion du matériel (accès au processeur, mémoire, périphériques),

Le contrôle des entrées/sorties (clavier, écran, etc.),

La gestion de la mémoire et des ressources,

Le multitâche si nécessaire, ou du moins l'organisation des tâches

Qu'est-ce que le BIOS ? En quoi participe-t-il à la portabilité d'un système ?

Basic Input Output System. Ensemble de fonctions permettant d'accéder aux périphériques de manière « basique ».

Bibliothèque de fonctions et de procédures d'interruption.

Présentez le mécanisme d'ordonnancement « tourniquet à priorités dynamique »

Ordonnancement FIFO préemptif à quantum de temps d'utilisation du processeur.

Attribution d'une priorité initiale à chaque tâche, recalculée en fonction de son activité.

Est-il adapté au cas d'un système temps réel ? Pourquoi ?

Oui car le programme fait beaucoup d'entrées sorties et se verra donc favorisé par rapport aux tâches qui monopolisent le processeur et dont la priorité sera revue à la baisse.

A quoi sert le mécanisme de segmentation ? Présentez le fonctionnement de la segmentation.

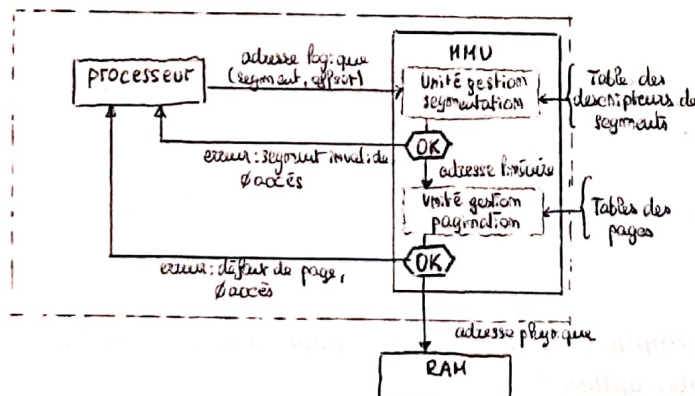
La segmentation répond aux contraintes de gestion de l'espace (découpage du programme en segment de taille variable) d'optimisation et de protection des informations.

Lors du chargement en mémoire du processus, la position de chaque segment en RAM est mémorisée dans la table des descripteurs de segments par le SE.

Les adresses contenues dans le programme exécutable sont sous la forme (numéro de segment, offset / début du segment). La recherche en RAM de chaque adresse est effectuée matériellement par la MMU.

Si cette adresse existe et que le processus a le droit d'y accéder, l'accès mémoire s'effectue. Dans le cas contraire, une erreur est générée.

Décrivez les étapes de la lecture par le processeur d'une donnée située en RAM dans le cadre de la segmentation.



Pourquoi est-il indispensable de tester le code retour de fopen(). Quel serait le comportement de votre programme si fopen() échoue et qu'aucun traitement d'erreur n'est effectué par le code ?

fd contiendra la valeur NULL (0) si fopen() échoue.

Le premier accès à la valeur pointée par fd reviendra donc à accéder à l'adresse 0 en RAM

-> « segmentation fault » et crash du programme

Qu'est-ce que la défragmentation d'un disque dur ? Expliquez en quoi elle permet d'améliorer les performances.

La défragmentation consiste à regrouper les fragments de fichier éparpillés sur le disque afin d'optimiser le temps d'accès du disque dur lors de la lecture de fichiers de taille importante. Ainsi la lecture par avance sera plus utile.

Ordonnancement ?

Gestion des files d'attente des tâches prêtes.

- Préemptif : ordonnanceur maître de la gestion de la ressource processeur. Peut reprendre la CPU à une tâche comme il veut.
- Non-préemptif : une tâche garde la gestion de la ressource processeur tant qu'elle n'est pas libérée volontairement.

Allocateur ?

Alloue le processeur dès qu'il est disponible à la tâche située en tête de la file d'attente des tâches prêtes. Gère la commutation de contexte

Qu'appelle-t-on la commutation de contexte ? Citez des événements pouvant déclencher une commutation de contexte.

Changement de tâche disposant du processeur. Au cours de la commutation de contexte, le contexte d'exécution est sauvegardé et celui de la nouvelle tâche est chargé à la place.

Accès à un périphérique, synchronisation, fin de quantum de temps

A quoi sert un SE ? Un ordinateur a-t-il obligatoirement besoin d'un SE ?

Lance l'exécution des programmes, gère leur déroulement en optimisant la gestion des ressources matérielles, interface utilisateur, fournit commandes et outils.

Aujourd'hui oui, sinon il faudrait indiquer à l'ordinateur tout ce qu'il devrait faire à chaque action entreprise – impossible !

Qu'est ce qui caractérise un système temps réel ?

Nécessite de garantir le respect de contraintes temporelles.

Principes de mécanismes de cache en illustrant par des exemples.

Mémoire cache : zone de mémoire intermédiaire entre deux composants de vitesse d'accès très différentes. Stockage temporaire d'informations pour éviter l'accès systématique au composant le plus lent.

Cache matériel entre le processeur et la RAM pour les données et les instructions

Cache logiciel des entrées-sorties pour les accès aux fichiers (entre RAM et contrôleur disque)

Cache matériel du contrôleur disque.

- Ce type de système vise à offrir une réactivité rapide
- Un système temps réel est conçu pour respecter des deadlines précises

En résumé, un système multitâche interactif privilégie la réactivité globale, tandis qu'un système temps réel garantit des temps de réponse déterministes pour des tâches critiques.

Quelle est la principale différence entre un système multi-tâche préemptif interactif et un système temps réel ?

Préemptif : le système peut décider de suspendre ou d'interrompre l'exécution d'une tâche, une tâche ne peut pas bloquer le système.

Temps réel : tâches activées par un événement extérieur.

Temps partagé : gère le partage du temps de travail du processeur entre les différentes tâches en cours.

Allocation mémoire à l'adresse 0. Lors de l'exécution, on a « Segmentation Fault ».

Pourquoi ?

Erreur de segmentation.

Décrivez les événements déclenchés par la frappe d'une touche au clavier dans un système où les entrées-sorties sont gérées par interruption.

Emission d'une interruption par le contrôleur clavier à destination du contrôleur d'interruption qui la transmet au processeur.

Ce dernier détermine l'origine de l'interruption (dialogue avec le contrôleur d'IT) dans la table des vecteurs d'interruption et l'exécute.

Cette fonction est chargée de récupérer la valeur de la touche frappée dans le registre de données du contrôleur clavier.

Que contient la table des vecteurs d'interruption ? Où est-elle stockée ? Quand et par qui est-elle renseignée ou mise à jour ?

Contient l'adresse en mémoire des procédures d'IT. Stockée au début de la RAM.

Au démarrage, chargée avec les adresses des fonctions du BIOS, puis modifiée par le système quand il est chargé, puis éventuellement en cours d'exécution par les programmes.

Décrivez les optimisations effectuées par les mécanismes de cache dans le cas d'une boucle de lecture du contenu d'un fichier.

Cache des entrées-sorties du système : demande de lecture au contrôleur disque de plus de bloc de données que demandés par le programme. Stockage de ces données dans la mémoire du système. Lors des demandes suivantes, il ne sera pas utile d'accéder au disque.

Cache disque : Même principe de lecture anticipée. Les données lues par avance sont stockées dans les registres du contrôleur de disque.

Permet à 2 niveaux de réduire le nombre d'accès au support physique qui est très lent.

Comment est assurée la protection d'un processus contre le code des autres processus ?

Précisez en particulier les rôles tenus par le matériel et le système.

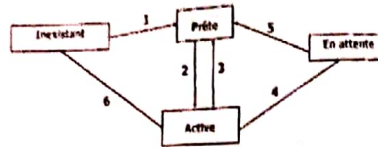
Par le mécanisme de segmentation qui permet d'affecter des droits d'accès à chaque segment d'un processus.

Le système d'exploitation renseigne la table des segments (localisation et droits) au chargement en mémoire d'un processus. Le matériel (MMU)...

Présentez rapidement le mécanisme de pagination.

La pagination est une technique de mémoire virtuelle qui permet d'utiliser le disque dur comme extension de la RAM. Chaque segment est divisé en page de taille prédéfinie qui peuvent être indifféremment stockées en RAM ou sur une mémoire de masse.

Dans un système multitâche, présentez le cycle de vie d'une tâche.



Prête : attend que le processeur lui soit affecté

En attente : La tâche est en attente d'un événement

Active : Le processeur exécute les instructions de la tâche en mode utilisateur ou superviseur

Inexistante : La tâche n'est pas encore créée ou son exécution est terminée. Elle n'est pas connue du système

- 1 – Chargement en mémoire d'une tâche par appel système d'un processus existant
- 2 – L'ordonnanceur choisit la tâche
- 3 – L'ordonnanceur choisit une autre tâche
- 4 – La tâche se met en attente d'un événement.
- 5 – L'événement attendu se produit
- 6 – Terminaison de l'exécution. Sortie des tables du système

Choix d'un buffer de 512 octets pour la lecture des données dans le fichier ?

512 oc = taille d'un secteur disque, unité minimale d'accès au disque dur.

Décrivez les 2 modes d'édition de liens. Quels sont les avantages et les inconvénients de chacun ? Le cache des entrées-sorties permet d'optimiser les accès en mode bloc. Décrivez ce mécanisme dans le cas des accès fichiers suivants :

- Écriture de données

Si les données sont dans le cache -> modification dans le cache -> marquage 'à écrire'
Sinon -> lecture physique des données -> stockage dans le cache -> lecture des données dans le cache -> modification dans le cache -> marquage 'à écrire'

- Lecture de données

Si les données sont dans le cache -> lecture dans le cache
Sinon -> lecture physique -> stockage dans le cache -> lecture dans le cache

A quelles contraintes de gestion de la mémoire doit répondre un système multitâches ?

Le système doit assurer la gestion et l'optimisation de leur stockage dans l'espace RAM disponible, la protection des données des processus contre le code des autres et permettre l'extension de l'espace RAM sur disque si nécessaire.

Différence appel système et appel fonction classique ?

L'appel système est un appel à un sous-programme intégré au système, tandis qu'un appel de fonction est un appel à un sous-programme du programme.