

TD n° 4 : Interruptions, dérouterments, appels svc

Notations :

- le contenu d'un mot d'état est représenté par $\langle \text{activité, mode, masquage, CO} \rangle$
- l'opérateur `adr` appliqué à un identificateur de variable ou de procédure désigne l'adresse de l'emplacement où est rangée la variable ou celle de la première instruction de la procédure.
- les notations `anc_svc`, `anc_dérout`, `anc_it_x`, `nouv_svc`, `nouv_dérout`, `nouv_it_x` désignent les emplacements de rangement et chargement des mots d'état.
- les appels au superviseur sont de la forme `svc` $\langle \text{code} \rangle$ où `code` désigne un numéro qui identifie la fonction demandée.
- un indicateur `cause` permet d'identifier la cause ayant provoqué un dérouterment.

Exercice 1. Le comptage du temps sur le PC est effectué via une interruption matérielle IRQ 0 du circuit 8259 (timer). IRQ 0 est déclenchée 18 fois par seconde.

1. Écrire, en langage algorithmique, une routine d'interruption de l'IRQ 0 qui permet d'afficher l'heure courante (en Heure, Minute, Seconde) sous la forme `HH:MM:SS`.

Indication : vous supposerez que toutes les variables utilisées sont correctement initialisées. Utilisez une variable pour compter le nombre d'interruptions.

2. Que se passe-t-il si l'exécution de la routine d'interruption dure plus de 0,06 seconde ?
3. Que peut-on proposer dans ce cas ?

Exercice 2. Calculer la taille de la mémoire

On cherche à écrire l'algorithme permettant de déterminer la taille réelle de la mémoire. Pour cela, nous allons utiliser le dérouterment pour adresse d'emplacement inexistant. La mémoire est constituée d'un nombre *nblocs* à déterminer de blocs de *p* mots. Le programme tente d'accéder au premier mot de chaque bloc. On fait l'hypothèse importante que la mémoire est contiguë. Ce programme fonctionne en mode maître (il est exécuté au démarrage du système).

Exercice 3. Délai de garde

Pour mesurer le temps, on utilise le plus souvent une horloge. Une horloge est un mécanisme qui envoie des impulsions périodiquement. Ces impulsions sont utilisées pour décrémenter un compteur contenu dans un registre spécial ou dans un mot réservé de la mémoire. Le passage de ce compteur à 0 déclenche une interruption.

L'exécution d'un processus *p* avec délai de garde consiste, après un délai *q*, dit délai de garde, à interrompre *p*. Une procédure spécifique `trait_erreur` est alors appelée. L'appel gardé de *p* est réalisé par un appel `svc` dont les paramètres sont *p*, le délai de garde *q* exprimé en périodes de l'horloge et le processus de reprise `trait_erreur`.

Écrire l'appel au superviseur qui réalise l'appel gardé et la procédure `it_horloge`.

Exercice 4. Gestion de travaux en temps partagé

Un ensemble de processus doit être exécuté sur une machine en utilisant le processeur en temps partagé : le processeur est successivement alloué aux processus dans un ordre fixe, par tranches de temps de durée fixe *q* dit quantum de temps. Si un processus se termine avant la fin du quantum, le processeur est immédiatement alloué au processus suivant.

Les processus sont numérotés de 0 à *n* - 1. On définit pour chaque processus *i* une zone de rangement `mep[i]` pour son mot d'état et une zone de rangement `reg[i]` pour ses registres. Ces zones servent à sauvegarder le contexte du processeur chaque fois qu'un processus est interrompu, pour permettre la reprise ultérieure du processus.

Les processus sont gérés dans une file circulaire notée `file_proc` munie de trois procédures

`entrer`, `sortir` et `suivant`.

1. Si on suppose que les processus ne terminent jamais, écrire la procédure d'initialisation et la procédure `it_horloge`.
2. Pour prendre en compte la terminaison des processus, chaque processus doit signaler au système qu'il se termine, au moyen d'un appel au superviseur `svc` (`fin`). Donner la portion du programme `traiter_svc` correspondant à la fin d'un processus.