Interruptions

Exercice 1

Nous appelons activité le phénomène résultant de l'exécution ininterrompue d'une procédure unique. Nous appelons contexte d'une activité l'ensemble des informations accessibles au processeur au cours de cette activité. Le contexte comprend un contexte de processeur (registres) et un contexte en mémoire (segment de texte, segment de donnée). Le passage d'une activité à une autre est réalisé par des instructions spéciales, l'appel et le retour de procédure, qui réalisent une commutation de contexte. Une procédure q (appelante) provoque l'exécution d'une procédure p (appelée) au moyen d'une séquence d'appel, qui comporte les étapes suivantes :

- préparation des paramètres transmis de q à p,
- sauvegarde d'une partie du contexte de q, devant être retrouvée au retour,
- remplacement du contexte de q par celui de p.

Le retour a un schéma à peu près symétrique :

- préparation des résultats transmis de p à q,
- restauration du contexte de q sauvegardé avant l'appel.

En faisant les hypothèses suivantes :

- les paramètres d'une procédure sont transmis par valeur ; un résultat unique est rendu au retour,
- les procédures peuvent être appelées récursivement

écrire les algorithmes d'appel et de retour de procédure gérant la pile d'exécution.

Exercice 2

On distingue trois mécanismes donnant lieu à des commutations de contexte :
- une interruption est déclenchée par une cause externe à l'instruction en cours ;
- un déroutement est déclenché par une anomalie dans l'exécution d'une instruction ;
- un appel au superviseur est une instruction particulière.

Le **mot d'état** du **processeur** (mep) est représenté par un enregistrement de la forme :

< activité, mode, masquage, compteur ordinal >, où
activité = attente ou active
mode = maître (système) ou esclave (utilisateur)
masquage = masqué (interruption bloquée) ou démasqué

Si **mep** désigne un mot d'état, ses champs sont désignés par mep.act, mep.mode, mep.masq et mep.CO. La notation **Mp[adr]** désigne le contenu de la mémoire à l'adresse adr.
Lors d'une commutation de contexte, le mot d'état courant est placé dans un endroit appelé **ancien_mep**, puis sa nouvelle valeur est chargée depuis l'emplacement **nouveau_mep**. Ce dernier contient notamment l'adresse du programme traitant de l'interruption. On dispose de plus d'une instruction **charger_mep(M)** dont l'effet est de charger le mot d'état M.
La notation **adr OBJ** désigne l'adresse de l'objet OBJ.

a) Mesure de la taille d'une mémoire par déroutement.

Un système d'exploitation est en général destiné à être utilisé sur diverses configurations d'une même machine. Le système doit être adapté à la configuration utilisée. La création d'une telle version spécifique est appelée génération du système. Pour réduire le nombre de générations, certains paramètres de la configuration, comme par exemple la quantité de mémoire présente dans l'ordinateur, sont automatiquement déterminés lors de l'initialisation du système.

On suppose que la mémoire est constituée de blocs de p mots, numérotés à partir de 0, et que l'accès à une adresse mémoire inexistante provoque un déroutement. Écrire un algorithme déterminant la taille mémoire d'un système (c'est-à-dire le nombre de blocs).

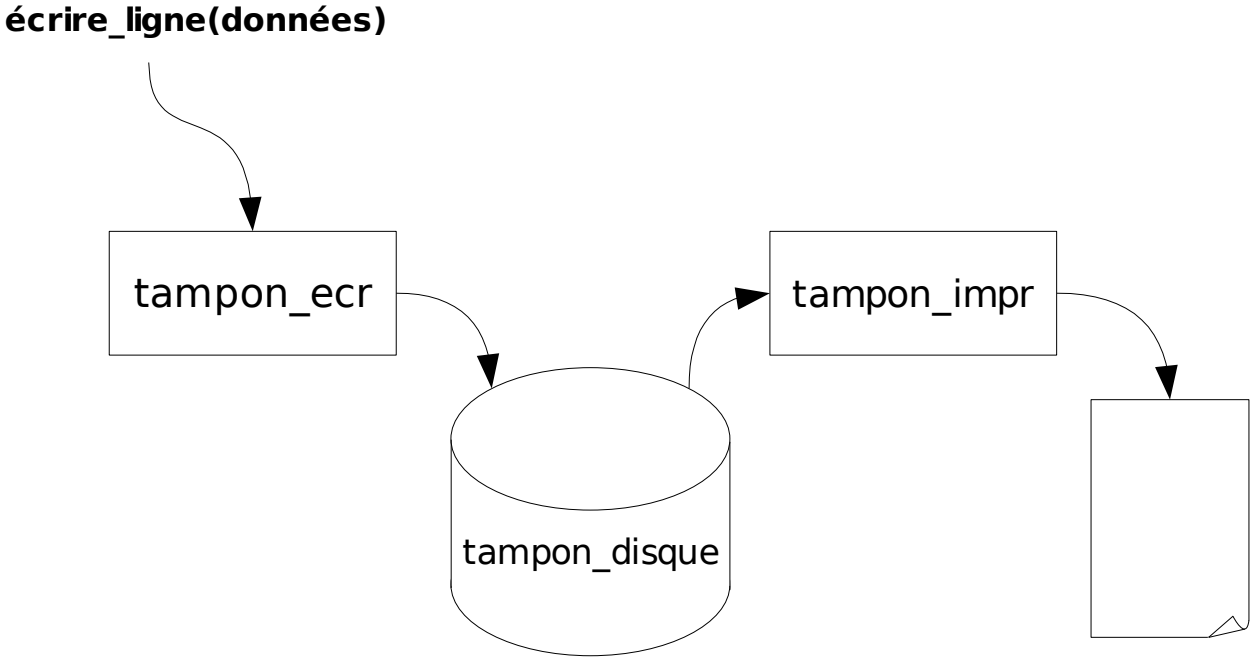
b) Simulation d'instructions manquantes.

On souhaite simuler un jeu d'instructions arithmétiques en virgule flottante sur une machine sans coprocesseur mathématique. Sans ce coprocesseur, l'exécution d'une instruction en virgule flottante (div, mult) par la machine provoque un déroutement pour instruction inexistante et l'arrêt du programme avec un message d'erreur.

Ecrire les procédures permettant de mettre en place une simulation logicielle de ces instructions en virgule flottante : lorsqu'un programme exécutera une telle instruction, une procédure simulant son exécution (rendant le même résultat) sera automatiquement appelée.

Exercice 3 Spoule d'imprimante

Les performances d'un système peuvent être améliorées en faisant transiter toutes les entrées-sorties lentes par des tampons. Pour un programme utilisant des périphériques lents, tout se passe comme si ces périphériques avaient un débit voisin de celui du disque. Dans le cas d'une imprimante, l'organisation du flot de sortie est schématisée par la figure suivante :



Ce mode de fonctionnement est appelé spoule, dérivé du terme anglais « spool » (Simultaneous Peripheral Operations On Line).

Les tampons mémoire sont des structures comprenant :
- un tableau **tab** de N cases de données (1 donnée par case) numérotées de 0 à N-1 ;
- 2 indices (tête et queue) permettant de gérer le tableau de façon circulaire, selon la stratégie FIFO.

La gestion des tampons est telle que :
- le premier transfert est initialisé par la première demande d'écriture,
- la fin d'un transfert d'une donnée (1, 2 ou 3) provoque une **interruption** de fin d'E/S, qui relance le transfert de la donnée suivante si possible.

Lorsqu'une interruption marque la fin d'un transfert, il faut relancer :
- le transfert suivant, si possible ;
- une activité antérieurement mise en attente parce que le tampon de départ était plein ;
- une activité antérieurement mise en attente parce que le tampon d'arrivée était vide ;

Les commandes permettant de gérer les E/S (DMA) disque et imprimante sont :

- **SIO <n, sens, périphérique, adresse mémoire, adresse périphérique>** pour lancer une requête d'E/S d'une donnée entre la mémoire et un périphérique

- **TIO <n>** pour tester un canal d'E/S (actif ou non actif)

où **n** représente le canal d'E/S (1,2 ou 3), **sens** est le sens du transfert par rapport au périphérique (lecture/écriture).

a) Donner les modes d'exécution d'une action de lecture/écriture dans un tampon mémoire.

b) Quelles sont les actions à réaliser sur le tampon_ecr
   - lors d'une demande *écrire_ligne*
   - lors d'une interruption de fin d'écriture disque (1) ?

c) Ecrire l'initialisation du système (mise en place du traitement du SVC et des interruptions de fin d'E/S), l'algorithme de la primitive *écrire_ligne* (exécutée par appel au superviseur) et l'algorithme de traitement des interruptions de fin d'E/S, en distinguant les 3 cas possibles de fin d'E/S (1, 2, ou 3).