# Piles et files

**Question 1** Expliquer sommairement le fonctionnement d'un gestionnaire des interruptions de tâches.

**Question 2** Comment fonctionne une pile LIFO (dernier entré, premier sorti) et les instructions associées push/pop ou empiler/dépiler ?
Comment fonctionne une file FIFO (premier entré, premier sorti) et les instructions associées enfiler/défiler ?

**Question 3** Les serveurs et gestionnaires d'impression fonctionnent comme des files : ils doivent traiter les requêtes dans l'ordre dans lequel elles arrivent. La mémorisation des pages web visitées par un navigateur se fait généralement dans une pile.
Pour chacune de ces deux structures de données, décrire sommairement un autre exemple d'utilisation pertinent.

**Question 4**
Le programme suivant, réalisé en assembleur , utilise les commandes `push`, `pop` et `mov` où `mov dest,src` copie le contenu de `src` en `dest`. Par exemple :

```
mov eax,[a]     ; copie la valeur de a dans le registre eax
```

Programme :

```
1 mov eax , 7
2 push eax
3 mov ebx , 9
4 mov ecx , 5
5 push ebx
6 push ecx
7 pop edx
8 pop eax
9 pop ebx
```

Tracer ce programme réalisé en assembleur en précisant l'effet de chaque ligne de commande.

**Question 5**

Ecrire un programme qui empile la suite des somme des premiers entiers : 1, 1+2, 1+2+3, 1+2+3+4
Outre les commndes citées ci-dessus, on utilise la commande add dont la sémantique est donnée ci-dessous :

```
1 mov eax , 7 ; eax=7
2 mov ebx , 5 ; ebx=5
3 add eax , ebx ; eax s e ra 12
```

Commentaires (en petits caractères) et réponses

---

**Question 1**

> Cette question de connaissance attendait une réponse centrée sur le thème *Piles et files*.

Le schéma de traitement d'une interruption est le suivant :

⋆ Interrompre l'exécution d'un programme en cours à la fin de l'instruction courante

⋆ Sauvegarder l'état du microprocesseur

⋆ Exécuter un sous programme dépendant de la nature de l'interruption

⋆ Restituter de l'état du microprocesseur avant l'interruption

Dans le cas de plusieurs interruptions successives, le traitement se fait dans l'ordre : **dernière interruption reçue = première interruption traitée**.

L'état du microprocesseur cité dans le schéma précédent est donc stocké dans une pile. Sa restitution se fait en reprennant le dernier état stocké.

> On pouvait ensuite ajouter des précisions sur le fonctionnement ou des améliorations, par exemple pour prendre en compte une hiérarchie de priorités, mais ce n'était pas l'objet de la question.

**Question 2**

Le principe des piles et des files est le stockage en mémoire de contenus de registres, avec la possibilité d'accéder exclusivement au dernier stocké dans le cas de la pile, et au premier dans le cas de la file. Le contenu concerné disparaît alors, rendant disponible celui qui a été stocké juste avant dans le cas de la pile et juste après dans le cas de la file.

• Empiler (ou push) stocke dans une pile et dépiler (ou pop) destocke (cas de la pile).

> `empiler eax`; stocke le contenu du registre eax en tête d'une pile par défaut
>
> `dépiler eax`; déstocke le contenu de la tête de la pile par défaut et le place dans eax.

Remarque : on peut envisager la commande pop sans opérande (le contenu est alors déstocké sans être transféré) mais pas la commande push sans opérande.

Dans la variante suivante, les commandes ont un opérande supplémentaire qui permet d'utiliser plusieurs piles :

> `empiler pil eax`; stocke le contenu du registre eax en tête de la pile pil
>
> `dépiler pil eax`; déstocke le contenu de la tête de la pile pil et le place dans eax.

• Enfiler et défiler sont les analogues pour la file.

Une pile est constituée d'une zone de mémoire dans laquelle on peut stocker temporairement des registres et d'un pointeur qui permet de repérer le sommet de la pile.

**Remarque sur l'analogie des piles d'assiettes :** *des trois versions données ci-dessous, deux correspondent à une organisation utilisable et une à une organisation inutilisable en informatique ; elles correspondent à des représentations mentales avérées dans les différentes copies (chacun cherchera sa chacune).*

version 1 : chaque assiette a une place en un endroit fixe et définitif (la place près de la fenêtre sur la table d'honneur, la place 3 sur la table 7, le dressoir près du chandelier ...). La pile est une liste (au sens naïf) de places d'assiettes. Le chef de rang peut ajouter une indication d'emplacement à cette liste (empiler) ou en enlever la dernière indication d'emplacement écrite (dépiler) et dans ce dernier cas verser le contenu de l'assiette qui s'y trouve dans une autre assiette (vide ou pas, mais dans ce dernier cas, son contenu initial est préallablement jeté à la poubelle) identifée par son emplacement. Les assiettes qui servent à stocker les mets pourraient être distinguées de celles que les convives peuvent utiliser : ce serait des plats.

version 2 : chaque assiette a un nom ou un emplacement. La pile est une rangée de plats sur un long dressoir et le chef de rang a placé devant l'une d'elle un objet-repère (la toque du chef par exemple). Le chef de rang peut verser le contenu d'une assiette de la salle dans le plat repéré du dressoir et déplacer le marqueur vers la gauche. Le chef de rang peut verser le contenu du plat repérée du dressoir dans la poubelle ou dans une assiette de la salle (vide ou pas, mais dans ce dernier cas, ...) et déplacer le marqueur vers la droite.

version 3 : chaque assiette a un nom (l'assiette au décor d'oiseaux de paradis, l'assiette en forme de coeur, ...). La pile est une rangée d'assiettes prises dans la salle et posée sur un long dressoir en une file (au sens naïf). Pour ajouter une assiette sur le dressoir, il faut donner son nom. On l'ajoute alors toujours du même côté (disons à gauche). Le chef de rang peut récupérer l'assiette du dressoir, toujours la plus à gauche, à condition qu'on donne son nom, pour l'apporter en salle.

Les deux premières versions correspondent à des réalités en informatique [a] et sont à peu près équivalentes dans leurs usages, et correspondent à des dispositions matérielles différentes. La troisième est totalement différente. Selon cette interprétation, dans le programme proposé à la question 4., la commande pop edx devrait : **(ne rien donner ou (exclusif) provoquer une interruption) ou (inclusif) provoquer un message d'erreur** suivant la façon dont le langage aurait été compilé. Remarquer que cette troisième version correspond à une organisation inutilisable pour la programmation car elle nécessite de connaître à tout moment le nom ou l'adresse du registre supérieur de la pile.

*a.* Une bonne étude de ces deux versions est le "Examen d'Architecture des Ordinateurs Majeure 1 – Polytechnique, lundi 10 Décembre 2001" disponible avec son corrigé à www-roc.inria.fr/ temam/teaching/x/exams/exam_01-02.pdf

**Question 3** Exemples de FILES

• *Lors de la gravure d'un cd, on utilise une file, ce qui permet d'avoir un flux de gravage constant.* [1]

• Les algorithmes de recherche en largeur d'un graphe dans un arbre utilisent une file pour mémoriser l'état de la recherche : la file contient la suite des noeuds à visiter. Voici l'idée :

Tant que le dernier noeud de la file a un fils, on enfile ce fils.
Dès que le dernier noeud de la file n'a plus de fils, on défile. [a]

*a.* Dans le cas d'un graphe, il faut en plus marquer les sommets visités

Exemples de PILES :

• *Une structure récursive fonctionne comme une pile. En effet, l'appel à l'élément précédent "stocke" en attente le résultat final : la demande de n appelle "n-1" qui arrive en tête de pile qui lui même appelle $n-2$. On empile donc jusqu'au rang initial 0. On peut alors traiter le rang 0 et on commence le dépilage jusqu'à la fin de la pile.* [2]

• Les algorithmes de recherche en profondeur dans un arbre utilisent une pile pour mémoriser l'état de la recherche : la pile contient la suite des noeuds de la branche en cours de visite. Voici l'idée :

Tant que le dernier noeud de la pile a un fils "non visité", on empile ce fils et on le marque "visité".
Dès que le dernier noeud de la pile n'a plus de fils "non visité", on dépile.

1. une copie : l'intéret d'un tel exemple est qu'il explique l'intéret de la file.
2. une autre copie

Exemples de FILES et PILES

• *Quand je vais au supermarché, je fais la queue en file à la caisse et si une nouvelle caisse ouvre, on dépile !*[3]

### Question 4

Ce programme n'était évidement pas conçu dans un but utilitaire ! Il échange les contenu des registres `eax` et `ebx` et transfère le contenu du registre `ecx` dans le registre `edx`. Par ailleurs, la sémantique des commande `pop` et `push` n'était pas donnée. Evidement, la réponse à la question revient à leur attribuer une sémantique et plusieurs étaient envisageables mais une seule (à ma connaissance), ne conduit à aucune contradiction :

• Considérer par exemple que la commande `pop eax` dépile dans la pile `eax` est contradictoire avec la sémantique imposée de la commande `move eax` qui nécessite que `eax` soit un registre.

• Considérer que la commande `push eax` empile le nom ou l'emplacement du registre `eax` rend nécessaire que la commande `pop eax` ait pour opérande le nom du dérnier registre empilé : il faudrait donc qu'à la rédaction de l'algorithme, l'analyste ait connaissance du registre qui a été empilé le dernier à tout instant. Par ailleurs, il n'y aurait pas d'opérande marquant l'utilisation qui serait faite du contenu du registre dépilé. Une telle sémantique n'est pas contradictoire, mais elle m'apparaît non opérationnelle.

• Ici, on utilise la sémantique proposée dans la réponse à la question 2.

1 `mov eax , 7` ; la valeur 7 est mise dans le registre eax
2 `push eax` ; la valeur de eax (qui est 7) est mise au sommet de la pile
3 `mov ebx , 9` ; la valeur 9 est mise dans le registre ebx
4 `mov ecx , 5` ; la valeur 5 est mise dans le registre ecx
5 `push ebx` ; la valeur de ebx (qui est 9) est mise au sommet de la pile
6 `push ecx` ; la valeur de ecx est mise au sommet de la pile la pile sera : 7 9 5
7 `pop edx` ; la valeur du sommet de la pile est mise dans le registre edx ;
edx sera 5 et la pile : 7 9
8 `pop eax` ; la valeur du sommet de la pile est mise dans le registre eax ;
eax sera 9 et la pile : 7
9 `pop ebx` ; la valeur du sommet de la pile est mise dans le registre ebx ;
ebx sera 7 et la pile vide

### Question 5

La première idée consite a écrire successievement 1, 3, 6, 10 dans une variable, puis à empiler :

```
1 mov eax , 1
2 push eax
3 mov eax , 3
4 push eax
5 mov eax , 6
6 push eax
7 mov eax , 10
8 push eax
```

3. une autre copie : l'intéret de cette réponse est de mettre en évidence qu'un même objet (ici des clients alignées) peut avoir plusieurs structure suivant les actions susceptibles de lui être appliquées : une liste d'adresses est une pile si on s'autorise les commandes empiler / dépiler mais peut devenir une file si on s'autorise les commandes enfiler / défiler ! De là à proposer les files comme exemple de piles... !

Que personne n'ait fait cette proposition, qui répond pourtant à la question, peut être analysé de deux manières différents :
• soit comme une illustration de la pertinence du concept didactique de contrat : en effet ce programme n'utilise pas la commande `add` mentionnéee dans la question ; dans ce cas, il suffisait d'ajouter la ligne inutile

```
9 add eax , ebx
```

• soit comme signe d'une compréhension profonde de la signification dynamique du travail de rédaction d'un algorithme : en effet, pour être généralisé à 3636, celui qui vient d'être proposé requiert le recopiage de chaque biligne et le calcul de la valeur transférée dans la variable doit être fait avant la rédaction du programme ; on chercher donc un algorithme susceptible de généralisation comme le suivant.

L'idée consiste à incrémenter la valeur contenue dans `eax` par la commande `add` :

```
1 mov eax , 0
2 mov ebx , 0
3 add eax , 1
4 add ebx , eax
5 push ebx
6 add eax , 1
7 add ebx , eax
8 push ebx
9 add eax , 1
10 add ebx , eax
11 push ebx
etc ...
```

Les trilignes sont recopiées sans changement. On voit alors comment une boucle permettrait de généraliser facilement à un entier quelconque, voire à un entier quelconque.

Enfin, il semblerait qu'un programmeur éthique et responsable vide ses piles à la fin des programmes [a]. Il faudrait alors ajouter des commandes `pop` .

a. comme un enseignant éthique et responsable efface son tableau en sortant de sa classe ?