

<p style="text-align: center;">info201 : Système d'exploitation Examen</p>
--

Pierre Hyvernât

Sébastien Monnet

UFR Sciences et Montagne, université Savoie Mont Blanc

Pierre.Hyvernât@univ-smb.fr

Sébastien.Monnet@univ-smb.fr

Durée : 1h30.

Un barème provisoire est donné dans la marge, un point négatif est réservé pour la présentation.

Partie 1 : architecture

(2) *Question 1.* Donnez, en justifiant, trois exemples de périphériques *autre que clavier ou écran* qui soient :

- un périphérique d'entrée,
- un périphérique de sortie,
- un périphérique d'entrée / sortie.

Partie 2 : langage d'assemblage

Des opérations arithmétiques du langage d'assemblage MMIX sont :

- ADD \$x, \$y, \$z pour additionner les registres \$y et \$z et mettre le résultat dans \$x,
- SUB \$x, \$y, \$z pour soustraire le registre \$z à \$y et mettre le résultat dans \$x,
- MUL \$x, \$y, \$z pour multiplier les registres \$y et \$z et mettre le résultat dans \$x,
- DIV \$x, \$y, \$z pour diviser le registre \$y par \$z et mettre le résultat dans \$x,
- CMP \$x, \$y, \$z pour mettre +1 ou -1 dans le registre X suivant que \$y est plus grand ou strictement plus petit que \$z,
- JMP n pour aller directement à l'instruction "+n". Lorsque n est positif, cela "saute" n-1 instructions.
- BP \$x, n pour aller directement à l'instruction "+n", mais *seulement lorsque \$x est strictement positif*. (Sinon, passe à l'instruction suivante.)

Les opérations ADD, SUB MUL et DIV ont une variante où le registre \$z est remplacé par une constante entière n entre 0 et 255.

(2) *Question 1.* En supposant que x et y sont deux registres contenant des entiers, comment peut-on effectuer l'opération " $x = (x + 2y + 3)^3 + x + y$ " ?

(3) *Question 2.* Comment pourrait-on coder une boucle for en MMIX ? Pour répondre, traduisez la boucle suivante en MMIX. Vous pouvez supposer que n est un registre contenant la valeur de n.

```
x = 0
for i in range(n):
    | x = x+1
```

Remarque : l'objectif est de coder une boucle. Le résultat de cette boucle particulière pourrait s'obtenir plus simplement avec MUL x, n, 1, mais ce n'est pas ce qui est demandé.

Attention, 1 point est réservé au fait que votre code marche lorsque n est négatif ou nul.

Partie 3 : shell et redirections

Rappels :

- on peut enchaîner les commandes `CMD1 | CMD2 | CMD3 ...` pour : lancer la commande `CMD1`, faire agir `CMD2` sur la sortie de `CMD1`, faire agir `CMD3` sur la sortie de `CMD2`, etc.
- La commande `ls -S` affiche les fichiers du répertoire courant par ordre décroissant de taille.
- La commande `head -n NB FICHER1 ... FICHIERn` (resp. `tail -n NB FICHER1 ... FICHIERn`) affiche les `NB` premières (resp. dernières) lignes de chaque fichier donné en argument. Si aucun fichier n'est donné, la commande agit sur les lignes de l'entrée standard.
- La commande `sort FICHER1 ... FICHIERn` affiche les lignes de tous les fichiers donnés en argument dans l'ordre alphabétique. Si aucun fichier n'est donné, la commande agit sur les lignes de l'entrée standard.

(1) *Question 1.* Donnez une commande qui donne les 5 fichiers les plus gros du répertoire courant.

(2) *Question 2.* Quelle sera la différence entre les 2 commandes

```
$ ls | sort | head -n 10
```

et

```
$ ls | head -n 10 | sort
```

(2) *Question 3.* Pour la même commande `CMD`, peut-il y avoir une différence de temps d'exécution entre `CMD | head -n 10` et `CMD | tail -n 10`? Justifiez votre réponse.

Partie 4 : fichiers

(2) *Question 1.* Donnez des motifs shell (en utilisant uniquement les jokers `*`, `?` et `[...]`) pour obtenir les listes de noms de fichiers / dossiers suivants :

- tous les noms commençant par la lettre `A`,
- tous les noms contenant exactement 5 caractères (extension comprise s'il y en a une),
- tous les noms avec l'extension `.jpg`, sans différencier minuscules et majuscules,
- tous les noms avec l'extension `.html`, dans un sous répertoire direct du dossier courant.

Dans le dernier cas, vous devrez donc capturer le fichier `./info201/tp1.html`, mais ni `./index.html` (qui n'est pas dans un sous répertoire) ni `info601/OLD/tp2.html` qui n'est pas dans un sous-répertoire *direct*.

(2) *Question 2.* Les systèmes de fichiers standard découpent les fichiers en *blocs*. Les données d'un fichier ne sont donc pas forcément consécutives. Expliquez, en donnant des exemples, ce que cela simplifie dans la gestion des fichiers.

Partie 5 : processus et ordonnancement

Rappel : l'ordonnanceur ne donne jamais le processeur à un processus bloqué, car il n'a pas d'instruction à exécuter.

(1) *Question 1.* Donnez une situation où processus bloqué se débloque et devient **prêt**.

(1) *Question 2.* Un processus interactif (par exemple, navigateur web) est-il plus ou moins souvent bloqué qu'un processus non interactif (par exemple, compression d'un gros fichier)? Justifiez votre réponse.

(2) *Question 3.* Est-il raisonnable de donner une priorité très élevée aux processus interactifs? Justifiez en comparant les situations où les processus interactifs ont une priorité très basse et celle où ils ont une priorité très haute.