

SR01 - TD 9

Exécution d'une commande lancée par le C

Ex1: Sortie d'un programme

créer un programme "**sortie1.c**" qui utilise l'appel système **exit()** pour passer une valeur de statut de terminaison au shell.

Cette valeur de sortie sera testée dans un **if** du shell (dans un script qui appelle "sortie1.c").

créer un programme "**sortie2.c**" qui fait un appel à la fonction **atexit(void (*fonc))** pour provoquer l'exécution de la fonction "**fonc**" juste avant la sortie du programme par **exit()**. Faites un test en faisant sortir le programme par **exit()** ou par envoi d'un "**control-C**".

Ex2: **system()** et **execv()**

Le but de cette partie est de faire exécuter les deux commandes "`/bin/ls -la`" et "`/bin/ls *.c`" à partir d'un programme en C.

Pour chacune de ces commandes, deux méthodes seront étudiées : utilisation de la fonction **system()** et utilisation de la fonction **execv()**.

A: Exécution avec la fonction **system()**

Écrire un programme "**exec1.c**" qui exécute successivement avec la fonction **system()** les deux commandes : "`/bin/ls -la`" et "`/bin/ls *.c`".

- Compilez et exécutez votre programme.
- Le résultat est-il satisfaisant ?

B: Exécution avec la fonction **execv()**

Complétez votre programme "**exec2.c**" en écrivant une fonction **my_system(char *path, char **argv)** exécutant la commande passée en paramètre en utilisant la fonction **execv()**. "path" représente le chemin de la commandes exécutée ("`/bin/ls`") et "argv" le tableau terminé par NULL des arguments passés à cette commande, le premier argument étant

le nom du programme lui-même. "path" et "argv" sont dans ce cas les arguments à passer à `execv()`.

Cependant, vous pouvez, si vous le désirez, utiliser toute autre fonction de la famille des fonctions `exec` à la place de la fonction `execv()` et passer à votre fonction `my_system()` les paramètres qui vous paraissent alors judicieux.

- Dans votre programme principal, appelez successivement votre fonction `my_system()` avec les paramètres permettant d'exécuter les commandes `" /bin/ls -la"` et `" /bin/ls *.c"`.
- Compilez et exécutez votre programme.

C: Constatations et correction du programme

- Que constatez-vous au sujet de l'exécution de `execv()` ?
- Modifiez votre fonction `my_system()`, dans `exec3.c`, pour contourner le comportement de `execv()` mis en évidence.
- Après recompilation, que constatez-vous lors de l'exécution de `" /bin/ls *.c"` avec `my_system()` ? Expliquez la raison du comportement constaté.
- Quelle commande faudrait-il exécuter pour que le `" /bin/ls *.c"` avec `my_system()` s'exécute correctement ? Commentez. Exécutez cette commande en utilisant `my_system()`.