

STI 3^{ème} année – Programmation Système

TD 1: Norme POSIX

C. Toinard – J.-F. Lalande - J. Briffaut

POSIX est le nom d'une famille de standards définie depuis 1988 par l'IEEE et formellement désignée IEEE 1003. Ces standards ont émergé d'un projet de standardisation des API des logiciels destinés à fonctionner sur des variantes du système d'exploitation UNIX.

Le terme POSIX a été suggéré par Richard Stallman en réponse à la demande de l'IEEE d'un nom facilement mémorisable. C'est un acronyme de **Portable Operating System Interface**, dont le X exprime l'héritage UNIX de l'Interface de programmation.

Voici les différentes versions de la norme POSIX :

Norme	Symbole	Valeur
POSIX.1-1988	_POSIX_SOURCE	
POSIX.1-1990 (1003.1)	_POSIX_C_SOURCE	1
POSIX.1b-1993	_POSIX_C_SOURCE	199309L
POSIX.1c-1996	_POSIX_C_SOURCE	199506L
POSIX.1-2001	_POSIX_C_SOURCE	200112L
XPG3	_XOPEN_SOURCE	1
XPG4	_XOPEN_SOURCE	4
XPG4	_XOPEN_VERSION	4
SUS	_XOPEN_SOURCE	1
SUS	_XOPEN_SOURCE_EXTENDED	4
SUSv2	_XOPEN_SOURCE	500

Pour compiler de manière conforme à POSIX.1 :

```
gcc -D_POSIX_C_SOURCE=1 -c source.c
```

1 Constantes de compatibilité

Exercice 1 Créez un fichier source C permettant de tester le positionnement des différentes constantes définissant la compatibilité POSIX au moyen de deux façons différentes.

Première façon :

```
#ifdef _POSIX_SOURCE
printf("_POSIX_SOURCE=%d\n", _POSIX_SOURCE);
#endif
```

Deuxième façon :

```
if (_POSIX_C_SOURCE == 2008)
printf("Code de 2008 inclu !\n");
else
printf("Code par défaut\n");
```

Exercice 2 Comment pourriez-vous tester votre fichier précédent au moyen du compilateur ?

Exercice 3 A votre avis, les tests précédents sont-ils réalisés à l'exécution ?

Exercice 4 Au moyen de la fonction `sysconf`, vérifiez la valeur d'autres constantes au moment de l'exécution.

2 Environnement

Les variables d'environnement sont des variables dynamiques utilisées par les différents processus d'un système d'exploitation

- L'environnement est accessible via : **`extern char **environ;`**
- Ou par la fonction POSIX :

```
#include <stdlib.h>
char *getenv(const char *nomDeVariable);
int putenv(const char *coupleNomValeur);
```

Exercice 5 Tapez "env" dans la console. Vous obtenez la liste de toutes les variables d'environnement du système.

Exercice 6 Ecrire un programme C équivalent à la commande env.

Exercice 7 Ecrire un programme C qui prend en argument le nom d'une variable d'environnement et affiche sa valeur.

3 Appel système et gestion des erreurs

Exercice 8 Ecrivez un programme permettant d'écrire le message hello sur la sortie standard à l'aide d'un appel système.

Exercice 9 Modifiez ce programme afin qu'il génère une erreur lors de l'appel système.

Exercice 10 Traitez l'erreur retournée au moyen de `perror` (1 ligne).

Exercice 11 Sans utiliser `perror`, traitez l'erreur retournée au moyen de `errno` et de `strerror`. Il s'agit d'écrire un équivalent de la fonction `perror` qui écrit le texte explicatif suivi d'un message spécifique à l'application.

4 Accès aux limites du système

Les limites du système ne dépendent pas forcément que du système. Dans le cas des fichiers par exemple, une limite peut dépendre du fichier que l'on considère.

Exercice 12 Etudiez la fonction `posix_pathconf`. Ecrivez une fonction **`void pr_pathconf(char *path, int name, char *message)`** qui prend en paramètre le nom de fichier à considérer, l'nom de la limite (entier) et le message textuel à afficher pour cette limite.

La fonction envoie sur la sortie standard le nom de la limite puis appelle la fonction `pathconf` afin de récupérer la valeur de la limite. La fonction envoie alors sur la sortie standard la valeur récupérée. Traitez les cas d'erreur avec `errno` de façon rudimentaire afin d'afficher le même message signalant l'erreur à la place du nom de la limite et de sa valeur.

Exercice 13 Créez un programme `main` permettant de tester votre fonctions. Votre `main` ne testera que les limites POSIX.1.

5 Base de données utilisateur

Exercice 14 Ecrivez un programme qui prend en paramètre un nom d'utilisateur et affiche le contenu de l'enregistrement correspondant de la base de données des comptes.

Exercice 15 Proposez une méthode pour le tester pour l'ensemble des comptes existants sur votre système.

Exercice 16 Comment se fait-il que vous ne trouviez pas votre login dans les utilisateurs systèmes alors que votre programme est capable d'afficher les informations relatives à votre login ?