TP UNIX N°3
INFO – 2° Année

Objectifs

Les fonctions de manipulation de fichiers et utilisation d'une zone de mémoire partagée.

Exercice 1 :

Grâce à un éditeur de texte ou un programme C créez un fichier contenant 5 lignes de 10 mots chacune. Puis écrivez un programme C capable d'ouvrir ce fichier texte (1er paramètre) et d'afficher les mots contenus dans une ligne (2ieme paramètre).

Pour cela, utilisez les fonctions open read et write.

Remarque

Les fichiers sur disque ne comportant aucune marque particulière de fin de fichier, le système détecte la fin du fichier lorsque la taille de l'offset a atteint la taille du fichier

Exercice 2 :

A l'aide des fonctions de bas niveau (open, read, write) créez un fichier ayant les permissions d'accès 0777, et contenant une chaîne de caractères quelconque (la table du code ASCII par exemple).

- avec la commande Unix : ls –l , observez le fichier créé (en particulier les droits), est-il conforme à ce que vous attendiez ? pourquoi ?
- avec la primitive : int stat(char *name, struct stat *buf) ; affichez les informations suivantes sur le fichier créé :
  - inode, mode, uid, gid, nombre de liens, taille, dates de dernier accès, dernière modification et créations.

Exercice 3 :

Nous avons vu avec les foncions de bas niveau: read et write comment manipuler un fichier. Les fonctions de haut niveau que sont: fopen, fscanf et fprintf, permettent un accès plus simple et formaté. En vous aidant des explications fournies sur ces fonctions et du programme de création ci-joint (programmation système sous UNIX, B. Coulange, pp67), analysez et commentez les résultats obtenus et réalisez le programme de lecture du fichier créé.

fopen

But: cette fonction permet d'ouvrir un fichier (en le créant éventuellement) associé avec des buffers et rend un "stream" permettant de l'utiliser ultérieurement avec des primitives de haut niveau.

Interface: #include <stdio.h>
FILE *fopen(nom,type)
char *nom, *type;

Description: Le fichier dont le nom terminé par un caractère nul est pointé par "nom" est ouvert et des buffers lui sont associés.

"type" est un pointeur sur une chaîne de caractères terminée par un caractère nul. Elle peut prendre une des valeurs suivantes:

"r" le fichier est ouvert en lecture.

"w" le fichier est créé et ouvert en écriture. S'il existait déjà, sa longueur est remise à zéro.

"a" le fichier est ouvert en écriture en fin de fichier s'il existe, il est créé sinon.

Valeur de retour: La valeur de retour de la fonction est un pointeur sur une structure de type FILE (décrit dans stdio.h). C'est le stream qui sera utilisable par la suite pour les accès au fichier. Si le fichier ne peut être accédé la valeur de retour est NULL et errno contient un code d'erreur.

Exercice 4 :

Ecrire un premier programme C permettant de créer une zone de mémoire partagée (SHM) puis créer deux programmes C permettant l'un d'écrire une chaîne de caractères dans cette SHM, l'autre permettant d'afficher le contenu de la SHM.

Primitives :

int shmget(key_t key, int size, int shmflg) ; permet soit de créer une SHM soit de récupérer un shmid pour une clef donnée (auquel cas size et shmflg ne sont pas utilisés).

*char *shmat(int shmid, char *shmaddr, int shmflg) ;* permet d'attacher la SHM identifiée par un shmid au segment de données du processus appelant.

*int shmdt(char * shmaddr) ;* permet de détacher la SHM du processus appelant qui ne pourra donc plus y accéder.