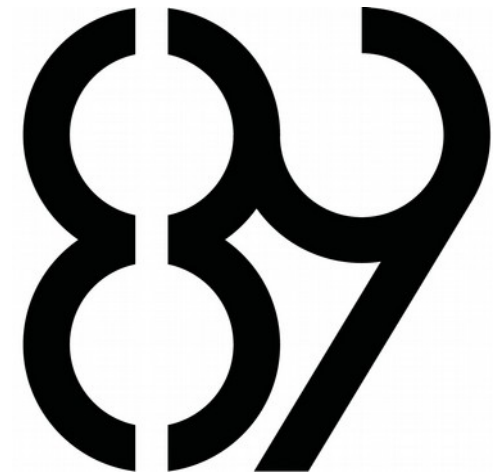




D A E M O N L A B



LS

Vous vous êtes beaucoup servit de cette commande.
Maintenant vous allez la faire.

- DaemonLab -
pedagogie@ecole-89.com

Ce document est strictement personnel et ne doit en aucun cas être diffusé.



INDEX

Avant-propos :

- 01 – Détails administratifs
- 02 – Propreté de votre rendu
- 03 – Règlement quant à la rédaction du code C
- 04 – Construction de votre rendu
- 05 – Fonctions interdites
- 06 – Projet



01 – Détails administratifs

Votre travail doit être envoyé via l'interface de ramassage de **l'Infosphère** :

Pour cette activité, vous rendrez votre travail sous la forme d'une archive au format tar.gz. Cette archive devra contenir l'ensemble de votre travail tel que demandé dans la section 4.

Pour créer cette archive .tar.gz, il vous suffit d'utiliser la commande suivante :

```
$> tar cvfz mon_fichier.tar.gz fichier1 fichier2 fichier3
```

Le nom « mon_fichier.tar.gz » étant à remplacer par le nom que vous souhaitez donner votre fichier archive, et « fichier1 », « fichier2 », « fichier3 » par les fichiers ou dossiers que vous souhaitez mettre dans cette archive. Vous pouvez vérifier le contenu de votre archive à l'aide de la commande « **tar -t mon_archive.tar.gz** ».

Ce travail est à effectuer seul. Vous pouvez bien sûr échanger avec vos camarades, néanmoins vous devez être l'auteur de votre travail. Utiliser le code d'un autre, c'est **tricher**. Et tricher annule **toutes** les médailles que vous avez reçu sur l'activité. La vérification de la triche est réalisée de la même manière que la correction : de manière **automatique**. Prenez garde si vous pensez pouvoir passer au travers.

Médailles accessibles :



Réussir à rendre :

Vous avez réussi à envoyer votre travail au système de correction.



02 – Propreté de votre rendu

Votre rendu, c'est à dire le contenu de l'archive ou du dépôt que vous entrez sur l'interface du TechnoCentre, doit respecter **strictement** l'ensemble des règles suivantes :

- Il ne doit contenir **aucun** fichier objet. (*.o)
- Il ne doit contenir **aucun** fichier tampon. (*.~, #*#)
- Il ne doit pas contenir votre production finale (programme ou bibliothèque)

La présence d'un fichier interdit mettra
immédiatement fin à votre évaluation.

Médailles accessibles :



Rendu propre

Votre rendu respecte les règles de
propretés imposées.



03 – Règlement quant à la rédaction du code C

Votre programme doit respecter la **Table des Normes**.



04 – Construction de votre rendu

Le programme de correction va construire une sous-partie déterminée de votre rendu afin d'effectuer des tests dessus. En voici les paramètres :

- Les fichiers qui seront compilés sont ceux qui auront l'extension *.c.
- Seuls les fichiers dans le(s) dossier(s) ./ seront compilés.
- Les fichiers seront compilés avec **-W -Wall -Werror**.
- Tous les fichiers seront compilés **ensemble**, cela signifie donc que chaque fonction doit être unique, et que vous pouvez utiliser les fonctions des autres exercices dans chaque exercice.

L'ensemble du code que vous rendez doit pouvoir être compilé.
En cas d'échec de la compilation, vous ne serez pas évalué.

Le nom de votre fichier exécutable sera ls.

Médailles accessibles :



Construction partielle

Les éléments requis de votre projet se construisent séparément.



05 – Fonctions interdites

Vous n'avez le droit à aucune autre fonction que celle précisée dans la liste ci-dessous :

opendir
readdir
closedir

stat
malloc
free

getpwuid
getgrgid

L'utilisation d'une fonction interdite est assimilée à de la triche.
La triche provoque l'arrêt de l'évaluation et la perte des médailles.



06 – Projet

Votre version du logiciel **ls** devra fonctionner de la même façon que la commande **ls -lah**. Vous devrez gérer une option, l'option **-S**, permettant de trier non pas alpha numériquement mais par taille de fichier. Votre programme sera alors équivalent à **ls -lahS**.

Ci-dessous, vous trouverez un exemple de code exploitant quelques unes fonctions dont vous avez besoin pour réaliser ce travail. N'oubliez pas de consulter les manuels des fonctions `opendir`, `readdir`, `closedir` et `stat`. Pour demander les manuels des appels systèmes, précisez **man 2**. Par exemple **man 2 stat**.

```
#include <dirent.h>
#include <unistd.h>

int main(void)
{
    DIR *dir ;
    struct dirent *browse;

    if ((dir = opendir("./")) == NULL)
        return (EXIT_FAILURE);
    while ((browse = readdir(dir)) != NULL)
    {
        puts(browse->d_name);
    }
    closedir(dir);
}
```