

```
int main( int argc , char* argv[] ) { ... }
```

TERMINAL

> ./app --count chat "chien bleu"

argc : 4 arguments sur la ligne de cmd.

argv : tableau de chaînes  
[ ] char\*

printf ("%s", argv[0]);

[1]

./app

--count

[2]

chat

[3]

chien, bleu



l'accès à argv[ ]



< argc

Comparaison chaîne de caractères:

#include <string.h>

i = strcmp("chien", "chat") i ≠ 0

i = strcmp("chocolat", "chocolat") i = 0

./app --count un deux trois ↵

argc=5

argv[0] = "./app"  
[1] = "--count"  
[2] = "un"  
[3] = "deux"  
[4] = "trois".

Tester si argc ≥ 2

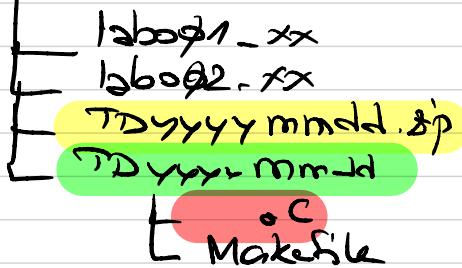
si strcmp(argv[1], "--count") == 0 VRai  
Alois { comptage ...

sinon  
si strcmp(argv[1], "--help") == 0 VRai  
Alois { usage .

;

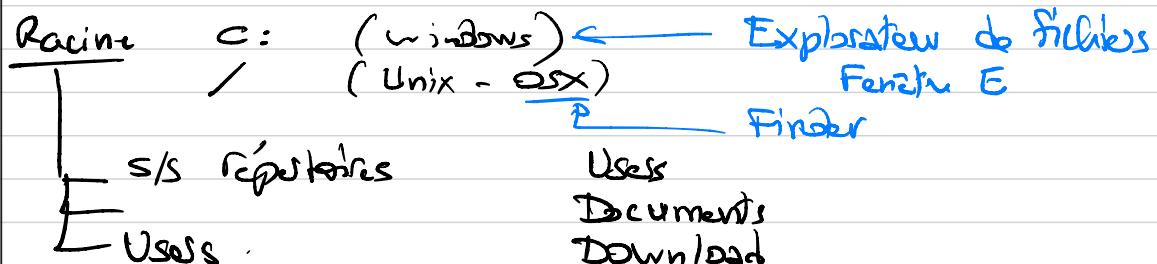
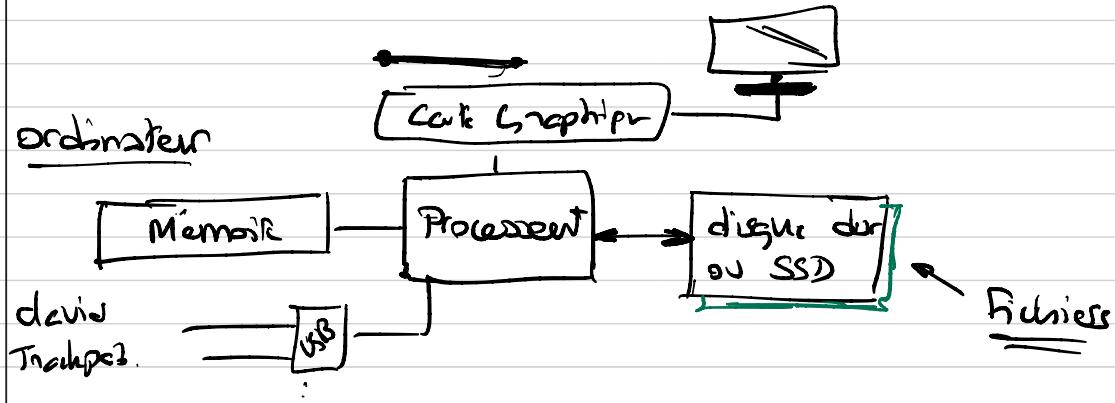
./app ↵

- ① Copier le fichier **TD yyyy mm dd.zip** à la racine du workspace
- ② Décompresser le fichier.

workspace

- ③ Renomme **le répertoire** en **TD 20210225**
- ④ **le fichier .c** et **TD20210225.c**

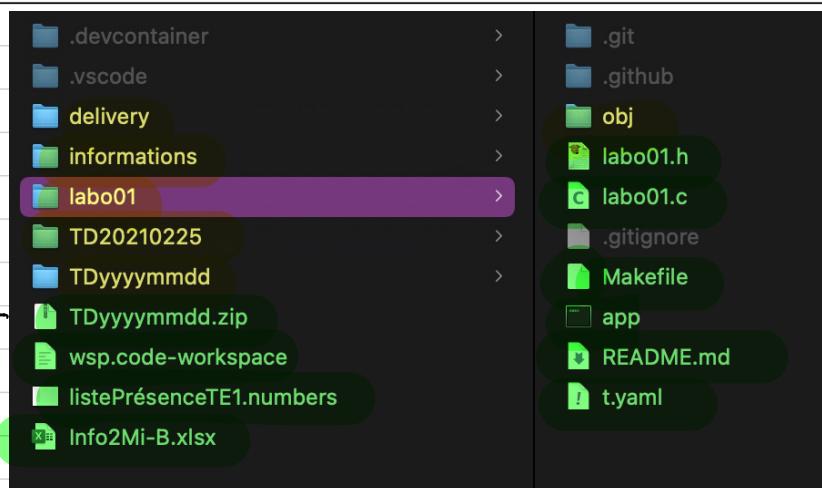
Dans le terminal : **cd** dans le répertoire **TD20210225**  
**make**  
**. /app** ⇒ **Hello World**



## Fichiers et vs Répertoire du workspace info2.

vs Répertoire

fichiers



Caractéristiques d'un fichier

labo01.c

.c

/users/pierre/documents/info2/lab01

← nom complet

← extension

← répertoire où est enregistré le fichier

Taille des fichiers

→ réelle.

→ effective.

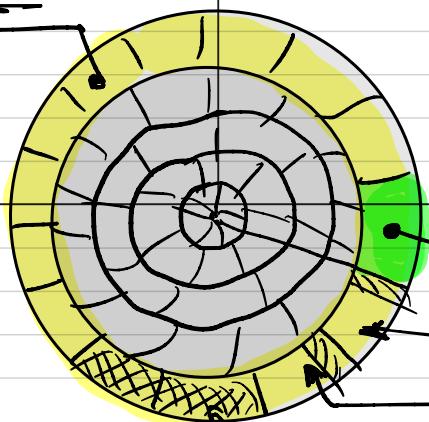
Terminal. ls -al

```
root@f280f120ea7a:/workspaces/Info2Mi-B/TD20210225# ls -la
total 32
drwxr-xr-x 10 root root 320 Feb 25 15:43 .
drwxr-xr-x 14 root root 448 Feb 25 15:40 ..
drwxr-xr-x 15 root root 480 Feb 22 11:31 .git
-rw-r--r-- 1 root root 430 Feb 22 11:28 .gitignore
-rw-r--r-- 1 root root 501 Feb 22 11:28 Makefile
-rw-r--r-- 1 root root 45 Feb 22 11:31 README.md
-rw-r--r-- 1 root root 292 Feb 25 15:43 TD20210225.c
-rwxr-xr-x 1 root root 10744 Feb 25 15:43 app
drwxr-xr-x 4 root root 128 Feb 25 15:43 obj
-rw-r--r-- 1 root root 9 Feb 25 14:18 text.txt
```

piste

taille  
réelle  
du fichier.

nom des fichiers



secteur, taille minimale effective d'un fichier = 1024 octets

p1.c 50 octets

p2.c 100 octets

p3.c 2000 octets → 2 secteurs → 200.

## Types de fichiers : TEXTE ou BINAIRES

**TEXTE** : humain compréhensible directement.

**BINAIRES** : nécessite un outil / programme pour comprendre le contenu.

x37.bmp

ext: bmp image bitmap

type binaire

travailler : - Visionneur image

- Aperçu

zz.mp3

ext: mp3 multiple compressé au format standard MP3.

type binaire

travailler : - Windows media player

- Musique

- Audacity



labo.c

ext: c texte au format C  
lisible directement.

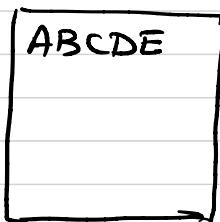
familles.xls

ext: xls tableau EXCEL.

type binaire

travailler : - EXCEL

test.txt



Lire

écrire

app

- 1) nom du fichier
- 2) emplacement
- 3) type d'accès → **lecture**  
**écriture**

⇒ **ouverture** : demande l'accès au fichier à travers l'osysteme.

5) **réponse si l'ouverture a été autorisée**.

Si Pas OK → erreur.

Si OK → accès au fichier.

6) ... opérations sur le fichier

⇒ **fonction** du fichier

TD 20210225: création d'un fichier txt.

1) nom = "text.txt" chaîne de caractères.  
2) emplacement: "/workspaces/info2/TD20210225/"  
   "."/ " ↗ à l'endroit où se trouve l'exécutable

ouverture d'effectuer sur: "./text.txt"

et nom 'w'

3) activité: création → mode d'écriture "w"

FILE \*fp = NULL;

fp = fopen ( nom , mode );

ex: fp = fopen ("./text.txt", "w");

{ l'app tente de ouvrir un fichier ./text.txt

↳ soit le fichier n'existe pas → il n'a donc rien.

↳ soit il existe déjà → il est effacé et remplacé.

↳ le contenu du l'ancien fichier est perdu.

⑤

Si ouverture impossible → fp vaut NULL.

sinon fp est utilisable pas la suite.

⑥ + tout...

⑦ fermeture

int ret; ret = fclose ( fp );

↳ si ret == 0 → fermeture correcte.  
sinon → problème.

1) créer le fichier text.txt

2) vérifier que création OK → ajouter msg de statut OK/NOK

3) fermer le fichier

4) vérifier que fermeture ok → ajouter msg de statut OK/NOK.