



INF2610

TP2



Chemin relatif et absolu

Un chemin relatif sera toujours donné **relatif à un répertoire en mémoire**. Un chemin absolu sera donné par rapport **au root de la mémoire (/)**.

Chemin **absolu**

```
/Users/benjaminboucher/Desktop/monFichier.txt
```

Chemin **relatif**

```
../../../../monFichier.txt
```

Un point « . » représente le répertoire courant et deux points « .. » vont représenter le répertoire parent



read

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

```
ssize_t read(int fd, void buf[.count], size_t count);
```

RETURN VALUE [top](#)

On success, the number of bytes read is returned (zero indicates end of file), and the file position is advanced by this number. It is not an error if this number is smaller than the number of bytes requested; this may happen for example because fewer bytes are actually available right now (maybe because we were close to end-of-file, or because we are reading from a pipe, or from a terminal), or because **read()** was interrupted by a signal. See also NOTES.

On error, -1 is returned, and *errno* is set to indicate the error. In this case, it is left unspecified whether the file position (if any) changes.

La valeur de retour de read lorsque la fonction lit «CTRL-D» est négative.



printf() VS write()

La fonction **write** effectue directement un appel système `write(fd, ..., ...)` qui **écrit directement** dans le fichier **représenté par le descripteur fd**

```
ssize_t write(int fd, const void *buf, size_t count);
```

La fonction **printf** écrit une série de caractères dans un buffer qui sera éventuellement vidé sur la **sortie standard** du processus appelant.

```
int printf(const char *format, ...)
```



Conditions pour vider le buffer

- Le buffer est est plein
- Contient un symbole de nouvelle ligne

```
printf("The buffer will be emptied after the following symbol : \n ");
```

- Le buffer est flushé (fflush)

```
int fflush(FILE *stream)
```



setvbuf

Pour résoudre cette question, référez-vous au lien suivant : <https://en.cppreference.com/w/c/io/setvbuf>

L'interface de **setvbuf**

```
int setvbuf(FILE* stream, char* buffer, int mode, size_t size);
```

