



Systèmes embarqués 2 – TP.10 – Mini Projet OS

Réalisation d'un système d'exploitation

Virtual File System – VFS

Le système de fichiers virtuels (Virtual File System – VFS) est un outil de modélisation puissant pour un système d'exploitation moderne. Il permet en effet non seulement de donner accès à des données stockées sur un disque physique, dans le cas de notre système la carte SD, mais il permet de représenter aussi bien des périphériques avec leurs différentes caractéristiques que des composants logiciels abstraits.

De manière classique, le système de fichiers permet de stocker sur un disque physique des données par l'intermédiaire de fichiers, lesquels seront placés dans une arborescence afin que l'utilisateur du système puisse structurer son information selon ses propres besoins.

Dans le cas d'un système de fichiers virtuels, le répertoire est utilisé pour représenter un périphérique ou un composant logiciel, tandis que les fichiers contenus dans le répertoire représentent les différents attributs ou caractéristiques du périphérique ou du composant logiciel. Si l'on prend par exemple la Click Board Thermo3 montée sur la cible, on peut très simplement la modéliser, soit :

thermo/	répertoire représentant le thermomètre
temp	fichier permettant de lire la température actuelle
thigh	fichier permettant de configurer la limite T-High du thermomètre
tlow	fichier permettant de configurer la limite T-Low du thermomètre
config	fichier donnant accès au registre de configuration du thermomètre

Il suffira finalement de placer ce composant « thermo » dans l'arborescence du système de fichiers virtuels à un endroit permettant de le retrouver facilement.

Arborescence

Pour cet OS, on proposera l'arborescence minimale suivante :

/	racine du système de fichiers virtuels (root)
dev/	point d'ancrage pour l'accès aux données de périphériques
proc/	point d'ancrage pour l'accès aux infos du noyau (optionnel)
sdcard/	point d'ancrage pour l'accès au système de fichier FAT de la carte SD
sys/	point d'ancrage pour l'accès aux données de statut et configuration des périphériques ou de composants logiciels du système

Services pour l'accès aux fichiers et aux répertoires

Trois interfaces proposent des services pour l'accès au système de fichiers et aux répertoires, soit FileIO, StandardIO et Dirent.

L'interface FileIO, via le fichier « unistd.h » propose des services selon la spécification « Single UNIX Specification, Version 2 ». FileIO est disponible sous tous les systèmes Linux. Elle propose des services sans cache des données au niveau de l'application, offrant ainsi un accès direct aux données des périphériques.



Systèmes embarqués 2 – TP.10 – Mini Projet OS

Réalisation d'un système d'exploitation

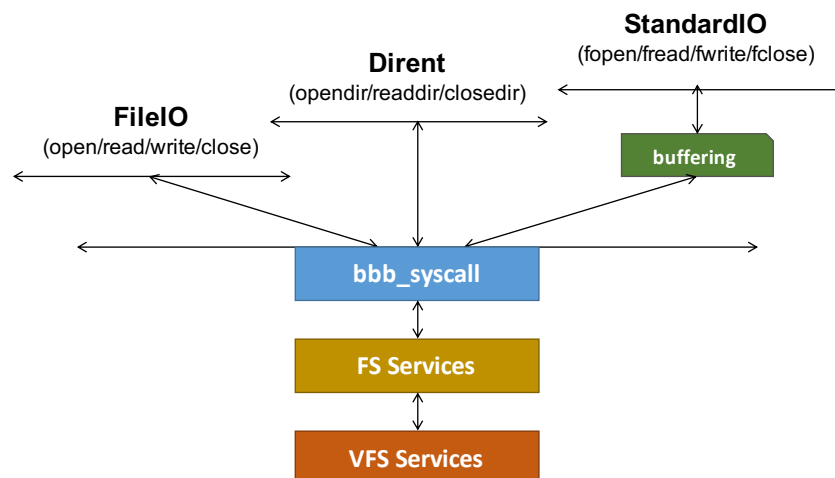
Virtual File System – VFS

L'interface StandardIO de l'ANSI C, via le fichier « `stdio.h` », propose des services selon la spécification IEEE 1003.1. StandardIO est disponible de manière standard sur toutes les implémentations du langage C. Elle propose des services avec cache des données au niveau de l'application.

L'interface Dirent propose des services pour traverser une arborescence et ainsi découvrir la structure de cette dernière. Elle permet également d'obtenir la liste des fichiers qu'elle contient.

Dans le cadre de nos laboratoires, les deux premières interfaces sont fournies et implémentées par la « Newlib ». La troisième interface, n'étant pas mis à disposition par la « Newlib », devra être réalisée séparément.

La « Newlib » est une bibliothèque C destinée généralement aux systèmes embarqués de petite taille et sans système d'exploitation. Cette bibliothèque est développée et maintenue par Red Hat. Elle est fournie en code source et peut très facilement être portée sur différentes plateformes en implémentant les services de bas niveau spécifiques au matériel. Dans le cas de notre infrastructure, ces derniers sont tous réalisés par l'intermédiaire du fichier « `bbb_syscall.c` ».



Les services proposés par l'interface FS offrent une façade pour l'accès aux services du système de fichiers virtuels et permettent ainsi de cacher la complexité de son implémentation.

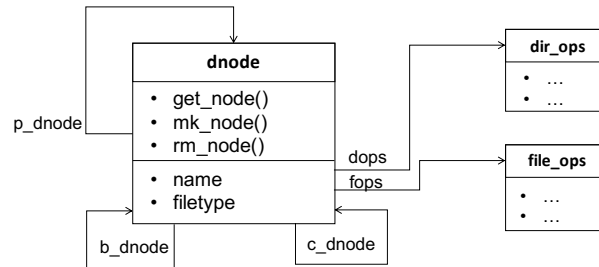
Structure élémentaire du VFS

Un système de fichiers virtuels est une arborescence de nœuds, lesquels représentent soit un répertoire soit un fichier. A chacun de ces nœuds on associera un nom (nom du répertoire ou nom du fichier), un type (répertoire ou fichier ordinaire) et des services permettant de manipuler les répertoires, les `dir_ops`, et des services permettant de traiter les fichiers, les `file_ops`.

Systèmes embarqués 2 – TP.10 – Mini Projet OS

Réalisation d'un système d'exploitation

Virtual File System – VFS

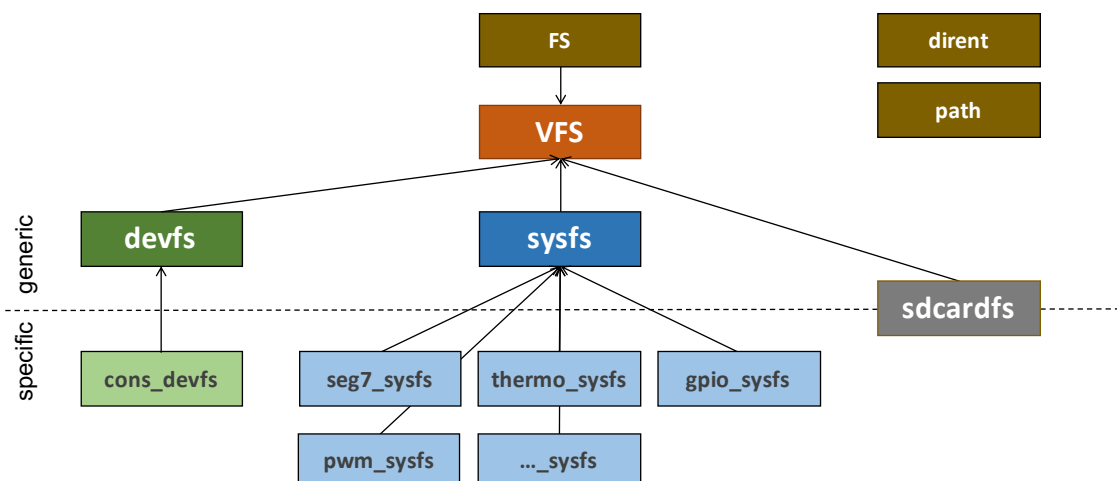


Cette structure relativement simple permet de modéliser un système de fichiers virtuels basic. Les deux références `b_node` (brother dnode) et `c_node` (child dnode) permettent de construire un arbre représentant la hiérarchie classique d'un système de fichier. La référence `p_dnode` (parent dnode) permet d'accéder facilement le nœud parent.

Trois services suffisent pour manipuler l'arborescence du système de fichiers virtuels. Le premier service `get_dnode` permet d'obtenir la référence sur le nœud correspondant au dernier nom du répertoire ou fichier souhaité dans l'arborescence. Le deuxième service `mk_dnode` permet de créer un nouveau nœud et de le placer dans l'arborescence. Quant au troisième service `rm_dnode`, il permet simplement d'extraire un nœud et de le détruire.

Diagramme des modules du VFS

Le diagramme ci-dessous représente les différents modules nécessaires à l'implémentation du système de fichiers virtuels.



Voici une description sommaire des modules principaux du système de fichiers virtuels :

- **vfs** : module implémentant les services de base du système de fichiers virtuels
- **devfs** : module générique permettant d'ajouter les services d'accès aux données d'un périphérique
- **sysfs** : module générique permettant d'ajouter des services d'accès aux configurations et statuts de périphériques ou composants logiciels abstraits
- **sdcardsfs** : module d'accès au système de fichier FAT de la carte SD (pas encore disponible)



Systèmes embarqués 2 – TP.10 – Mini Projet OS

Réalisation d'un système d'exploitation
Virtual File System – VFS

Le module `dirent` implémente les services pour traverser une arborescence. Le module `path` fournit des services pour manipuler les strings correspondant à des noms de fichiers ou de répertoires.

Les modules placés sous la ligne pointillée sont des modules spécifiques à la carte. Ils pourront/devront encore être développés afin de publier les fichiers d'accès aux périphériques de la carte Beaglebone Black utilisée dans le cadre des laboratoires de systèmes embarqués 1 et 2.

Travail à réaliser

Dans le cadre de ce laboratoire, les étudiant-e-s devront développer un système de fichiers virtuels, ceci comprend :

- Réaliser les trois services de la structure `dnode` (`get_dnode`, `mk_dnode` et `rm_dnode`)
- Concevoir l'interface public du module VFS
- Concevoir et réaliser l'un ou l'autre des modules spécifiques à la carte
- Valider l'implémentation

Conditions d'exécution

Pour télécharger le squelette du projet du dépôt centralisé :

```
$ cd ~/workspace/se12/tp
$ git pull upstream master
```

Pour mettre à jour les paths des includes dans eclipse :

- Ouvrir [Properties](#) pour votre projet
- Aller [C/C++ General](#) → [Paths and Symbols](#)
- Ouvrir [Includes](#) → [GNU C](#)
- Ajouter [/home/lmi/workspace/se12/bbb/source](#)

Le code et le rapport seront rendus au travers du dépôt Git centralisé :

- Sources : `.../tp/tp.10`
- Rapport : `.../tp/tp.10/doc/report_vfs.pdf`

Délai

- Le journal et le code doivent être rendus au plus tard 6 jours après le TP à 23h59