

1 U-boot

1.1 Compilation

On configure avec `make uboot-menuconfig` puis on effectue la compilation avec une des deux manières :

1. `make uboot-rebuild`
2. supprimer les fichiers puis `make`

La configuration de u-boot est stockée dans `.config`

1.1.1 Amélioration

Il est possible d'utiliser l'argument `-fstack-protector-all` pour ajouter des vérifications contre les buffer overflows (ou autres attaques sur le stack). Dans ce cas, un canary.

Si le canary est écrasé lors de l'exécution d'un morceau de code. On sais qu'il y a eut un dépassement dans le stack.

1.2 Démarrage

Si on appuie sur une touche, on entre en mode u-boot. La commande `booti` permet de lancer l'image linux (`boot` tout court va aussi lancer l'image Linux). Il existe aussi `bootz` pour charger une image compressée et `bootm` pour une image fit

Avec les commandes présentes dans `boot.cmd`, on indique l'emplacement dans la ram de `Image` et `nanopi-neo-plus.dtb`

Lors du démarrage, le Secondary Program Loader (`sunxi-spl`) va charger le fichier `u-boot.itb`

1.3 FDT (Flattened Device-Tree)

Le FDT contient une description hardware du système utilisée par Linux pour sa configuration. le FDT utilise deux fichiers :

- `.dts` : Device Tree Source (fichier ascii)
- `.dtb` : Device Tree Blob (fichier binaire)

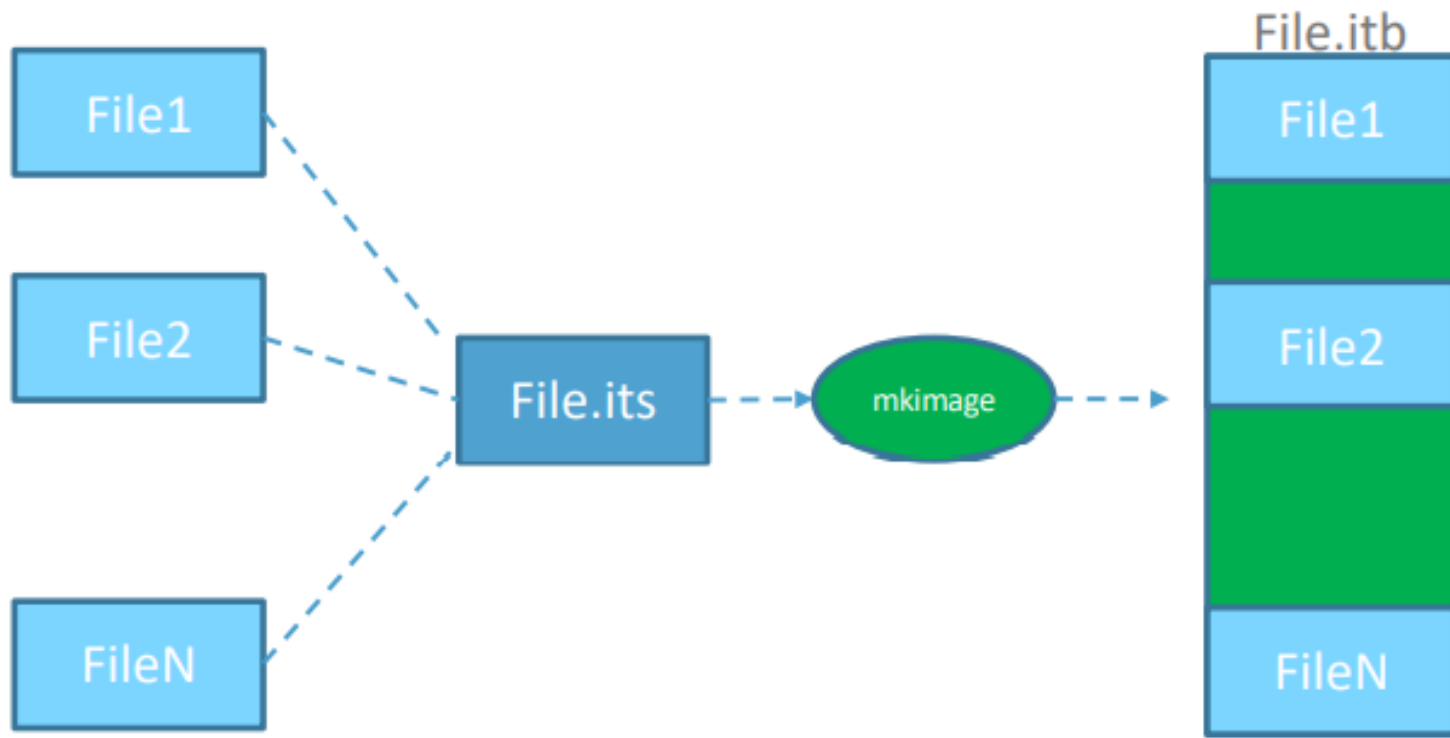
Possibilité de passer de `.dts` à `.dtb` avec la commande `dtc`.

U-boot utilise le fichier `sun50i-h5-nanopi-neo-plus2.dts` pour configurer Linux sur le NanoPi (information sur le port série, le processeur, etc...).

Le FDT est stocké dans le fichier `u-boot.itb`

1.4 FIT (Flattened Image Tree)

Nouveau format qui permet d'insérer plusieurs fichiers dans un seul



La commande `mkimage` permet de convertir un fichier `.its` en un fichier `.itb`.
Le fichier `u-boot.itb` est construit avec la commande

```
mkimage -f u-boot.its -E u-boot.itb
```