

Écriture de drivers pour Linux

Christophe BLAESS

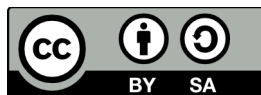
christophe.blaess@logilin.fr
<https://www.blaess.fr/christophe/>
<https://www.linkedin.com/in/christophe-blaess/>



Ingénierie et formations sur Linux et les logiciels libres
<https://www.logilin.fr>

Avant-propos

Ce support de formation est distribué sous licence **Creative Commons 4.0**



(Attribution - Partage dans les mêmes conditions).

Vous êtes libres de copier et partager ce document, en mentionnant son origine. Si vous l'intégrez dans un contenu plus vaste, ce dernier devra être distribué avec les mêmes droits.

Ce cours a été rédigé en utilisant des logiciels libres sur système d'exploitation Linux :

- *LibreOffice Writer* pour le support et la mise en page
- *LibreOffice Draw* et *Excalidraw* (en-ligne) pour les dessins vectoriels
- *Gimp* pour les images bitmap

« Linux » est un nom déposé par Linus Torvalds et administré par le « Linux Mark Institute ».

Logo « Pingouin mécano » dessiné par BeletteArtist (<https://www.instagram.com/beletteartist>)

ILD v. 9.12

2001-2025 Christophe Blaess

Introduction

Ce cours traite de la **programmation dans le noyau Linux** et de l'**écriture de drivers**.

Il s'agit d'une thématique avancée nécessitant une bonne familiarité avec le système Linux et une certaine aisance dans la lecture de programme en langage C.

Les démonstrations et **travaux pratiques** se dérouleront sur PC x86 et sur cartes **BeagleBone Black** (processeur ARM).

Le **support de cours** en version PDF est disponible ici :

<https://www.logilin.fr/files/support-ILD.zip>



Les **fichiers sources** des exemples sont disponibles dans le dépôt Github de Logilin :

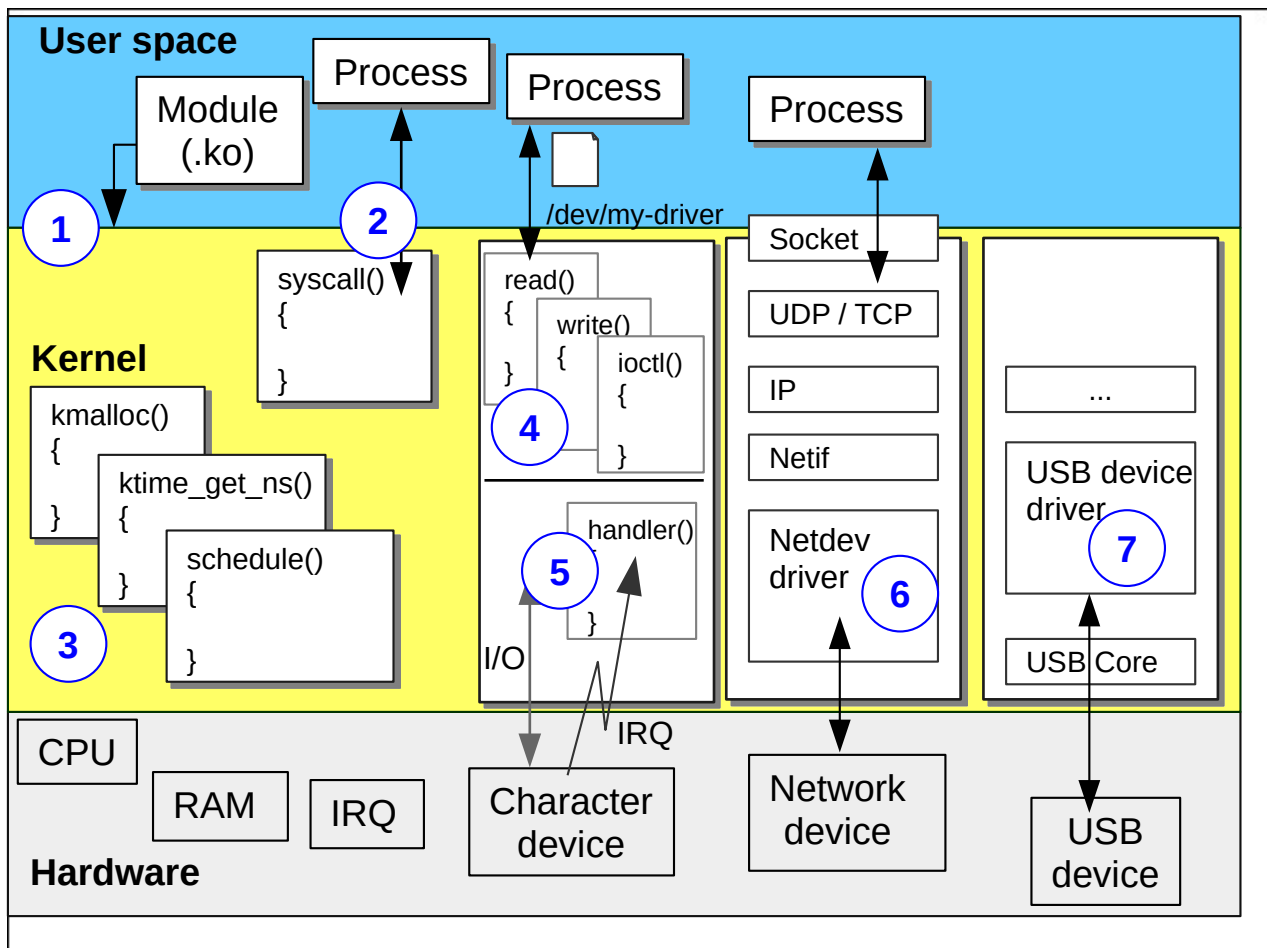
<https://github.com/logilin/ild>

Les **travaux pratiques** (énoncés d'exercices et solutions) se trouvent ici :

<https://www.logilin.fr/tp/ild>

Préconisations d'installation

- PC sous Linux, distribution courante (Ubuntu, Debian, Fedora, CentOS, etc.) assez récente.
- Accès internet indispensable
- Accès au compte *root* (ou *sudo*) indispensable.



Le noyau Linux et ses modules
<p>Noyau Linux et modules : introduction, évolutions du noyau, modules.</p> <p>Développement en mode noyau : outils, compilation du noyau, programmation noyau.</p> <p>Écriture de modules : <i>headers</i>, macros, licence messages, dépendances, <i>Makefile</i>.</p>
Les appels-système Linux
<p>Appels-système : principe, suivi d'un appel-système, préemptibilité du noyau.</p> <p>Environnement du noyau : contexte de tâche, espaces d'adressage.</p> <p>Système de fichiers /proc : gestion des entrées, <i>callbacks</i> de lecture et d'écriture.</p>
A.P.I. interne du noyau Linux
<p>Éléments de programmation noyau : chaînes, fonctions numériques.</p> <p>Mise au point et débogage : avertissements, paniques.</p> <p>Éléments temporels : <i>ticks</i>, mesure du temps, attentes, actions différées.</p> <p>Gestion mémoire : allocations, gestion des pages.</p>

Driver en mode caractère
<p>Enregistrement d'un driver : principe, numéros, modèles de drivers, classes.</p> <p>Méthodes fondamentales : ouverture et fermeture, lecture, écriture, paramétrage.</p> <p>Synchronisation entre appels-système : nécessité, <i>mutex</i>.</p>
Communication avec le matériel
<p>Entrées-sorties : ports, projections, <i>GPIO</i>, contextes du noyau</p> <p>Interruption : contextes, gestionnaires, traitements différés, synchronisation, attentes</p> <p>Communication par la mémoire : projections, <i>DMA</i>.</p>
Driver réseau
<p>Interfaces réseau : interfaces et protocoles réseau, sous-système <i>net_device</i>, administration.</p> <p>Driver réseau : enregistrement, activation, <i>socket buffers</i>, statistiques.</p>

Utilisation du bus USB
<p>Le sous-système USB : principe, interface USB de Linux ;</p> <p>Driver USB <i>Interrupt</i> : enregistrement du driver, <i>endpoints</i>, classes, <i>URB</i>.</p> <p>Aperçu de drivers pour autres modes : <i>endpoints Bulk</i> et <i>control</i>.</p>
Annexes
<p>Bibliographie</p> <p>Glossaire</p>

Travaux pratiques

Installation des fichiers sources des exemples de travaux pratiques

Dans un terminal, saisir :

```
$ git clone https://github.com/logilin/ild
```

```
$ cd ild/
```

```
$ git checkout 9.12
```

La première commande *git* télécharge les fichiers sources depuis le dépôt de Logilin sur Github.

La seconde sélectionne une version en adéquation avec ce support de cours.