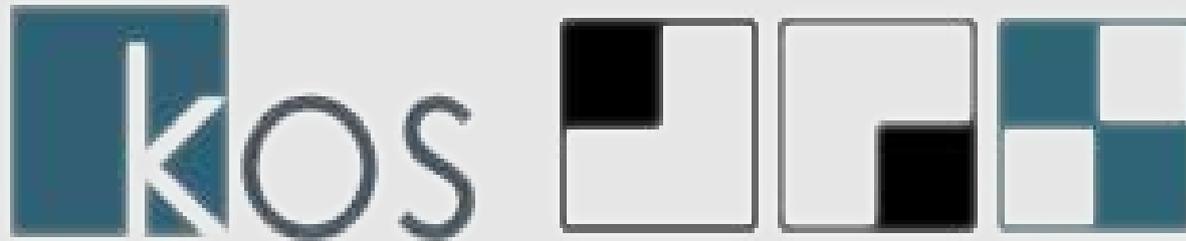


# KOS - Kid Operating System



Thomas Petazzoni

27 mai 2004 – 1er juin 2004

- Lancé en juin 1998
- Développé « from scratch »
- Jusqu'à 10 développeurs
- A l'heure actuelle, 3 développeurs actifs
  - David Decotigny
  - Julien Munier
  - Thomas Petazzoni
- **<http://kos.enix.org>**

- **Pour les développeurs : apprendre**
  - à programmer
  - à travailler de manière collaborative et distribuée
  - comprendre l'architecture x86
  - comprendre le fonctionnement d'un OS
  - debuggage et tests
  - **s'amuser**
- **Pour les autres**
  - Bibliographie
  - Documentations
  - Code source (GPL)
- Aucune application pratique prévue

## **Typique d'un projet Logiciel Libre :**

- CVS
- Web
- Mailing lists
- Rencontres physiques

## **Méthode « on the fly »**

- Conception pour le court terme
- Essai d'avoir quelque chose de satisfaisant d'un point de vue implémentation
- Améliorations itératives

# KOS : un système modulaire

- Noyau de KOS découpé en modules, jusqu'au coeur
- 25-30 modules
  - scheduler
  - vmm
  - pmm
  - arch/mm
  - arch/task
  - klavier
  - fs/devfs
  - fs/fat
  - ...
- Modules reliés au boot
- **Avantages**
  - Interfaces claires entre les sous-systèmes
  - Séparation code portable / non-portable
  - Chargement/déchargement (non implémenté)

# KOS : exemple de module

```
#include <loader/mod.h>

int hello_word(void)
{
    printk(``Hello World'');
    return 0;
}

__init_text static int
post_init_module_level3 (kernel_parameter_t *kp)
{
    UNUSED(kp);
    printk(``Hello World init ... Ok'');
    return 0;
}

DECLARE_INIT_SYMBOL(post_init_module_level3,
                    POST_INIT_LEVEL3);
EXPORT_FUNCTION(hello_word);
```

# KOS : interface utilisateur/noyau

- Fonctionnalité originale pour la gestion des ressources :

## **Karm**

- Remplacement du **ioctl** Unix
  - `int ioctl(int d, int request, ...);`
- Ouverture d'une ressource selon une interface
  - `int open(char *path, unsigned interface);`
- Appel système
  - Numéro de ressource
  - Numéro de méthode
  - Paramètres

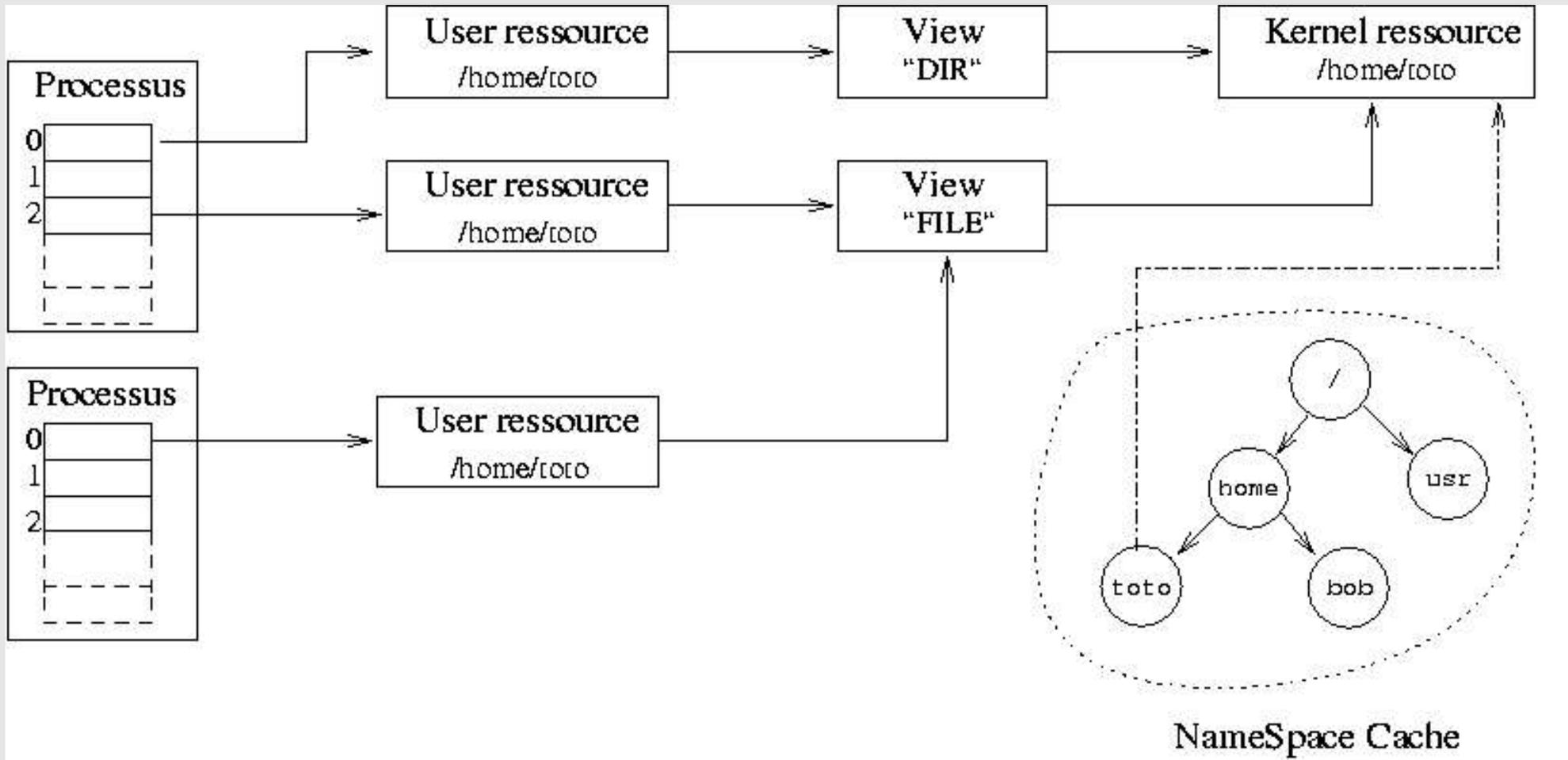
# KOS : définition des interfaces

Interfaces définies en XML pour générer :

- définitions coté noyau
- définitions et stubs coté utilisateur

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<interface name="block">
  <method name="read">
    <arg type="struct ures*" name="ur"/>
    <arg type="char*" name="buffer"/>
    <arg type="count_t" name="block_start"/>
    <arg type="count_t*" name="inout_block_count"/>
  </method>
  <method name="write">
    <arg type="struct ures*" name="ur"/>
    <arg type="const char*" name="buffer"/>
    <arg type="count_t" name="block_start"/>
    <arg type="count_t*" name="inout_block_count"/>
  </method>
</interface>
```

# KOS : fonctionnement général



Bibliothèques utiles : libcharfile, libblockfile, libfilemap.

- Système modulaire (chargement par Grub + loader)
- Gestion mémoire physique
- Gestion mapping mémoire
- Gestion mémoire virtuelle
- Allocateur mémoire pour le noyau
- Gestion des teams, threads noyaux et utilisateur
- Système karm avec appel système
- Pilotes de périphériques : disque, partition, console, clavier, série
- Systèmes de fichiers : FAT + devfs
- Gestion des interruptions
- Primitives de synchronisation pour le noyau
- Chargeur ELF
- Outils de débogage

## **Appels systèmes disponibles**

- fork
- exec
- brk
- getpid
- getppid
- open
- close
- read/write minimaux

## **Résultat :**

- Une mini bibliothèque C faite maison
- Un programme ELF chargé depuis le disque :
  - ouvre un fichier
  - se fork()
  - exécute un autre programme
  - créé des threads utilisateurs
  - alloue de la mémoire sur le tas

- Revoir toute la synchronisation dans le noyau
- Nouveaux appels systèmes
- Portage GNU libc

- KOS. **Kid Operating System**  
<http://kos.enix.org>
- Tigran Aivazian. **Linux Kernel 2.4 Internals.**  
<http://www.mc.man.ac.uk/LDP/LDP/lki/lki.html>
- Andries Brouwer. **A small trail through the Linux kernel**  
<http://www.win.tue.nl/~aeb/linux/vfs/trail.html>
- Jonathan Corbet. **Porting device drivers to the 2.5 kernel**  
<http://lwn.net/Articles/driver-porting/>
- Mel Gorman. **Understanding the Linux Virtual Memory Manager**  
<http://www.csn.ul.ie/~mel/projects/vm/>
- Hans-Peter Messmer. **The Indispensable PC Hardware Book**  
Number ISBN 0201403994, Addison-Wesley
- Alessandro Rubini and Jonathan Corbet. **Linux Device Drivers**  
<http://www.xml.com/lld/chapter/book/index.html>
- Andrew Tanenbaum. **Systemes d'exploitation.**  
Number ISBN 2100045547. InterEditions / Prentice Hall
- Uresh Vahalia. **UNIX Internals : The New Frontiers.**  
Number ISSN 0131019082. Prentice Hall.

**Inutile de noter : <http://thomas.enix.org/pub/conf/libreast2004/>**