Sos

A step-by-step implementation of a “do it yourself” Unix-like OS

David Decotigny and Thomas Petazzoni

Libre Software Meeting, July 2005

Contents

- Background
- Project
- Technical details
- Current status
- Future work

Background

- Development of the Kos operating system since 1998
- Double educational goal :
  - Learning software development, system programming and operating system internals – *for the developers* ;
  - Documentation about operating system development – *for others* ;
- First goal reached, but the second one was forgotten

Project

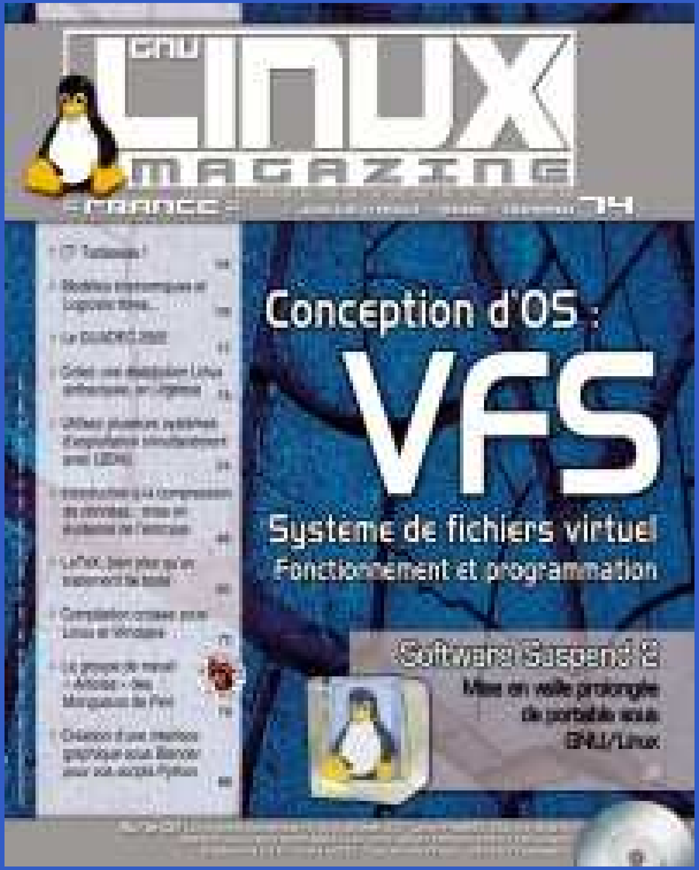
- Document the step-by-step implementation of a simple operating system ;
- Each month, some concepts are studied, described and implemented ;
- Simplest possible implementation, without hiding technical subtelties ;
- Base for OS education, or valuable source of inspiration for other OS ;
- **Not an original OS, only an original approach.**

Project

- Articles published almost monthly in the french *GNU/Linux Magazine* ;
- Articles available for free on the Sos website two months after their publication ;
- Mailing-list that hosts discussions about the Sos project.

GNU/Linux Magazine

GNU LINUX MAGAZINE
FRANCE
Créez votre OS
principes et implémentation
POV-Ray
Vers un rendu plus réaliste grâce aux caustiques et aux photons


GNU LINUX MAGAZINE
FRANCE
74
Conception d'OS :
VFS
Système de fichiers virtuel
Fonctionnement et programmation
Software Suspend 2
Mise en veille prolongée de portable sous GNU/Linux

Technical details 1/2

- IA32 architecture ;
- GNU compilation toolchain ;
- Boot via *Grub* or custom bootsector ;
- Monolithic kernel, interruptible but not preemptible ;
- Preemptive multitasked scheduling with kernel and user threads ;
- Address space separation, and notion of process ;

Technical details 2/2

- A *Virtual Filesystem* layer ;
- Drivers: keyboard, video, IDE, network, FAT filesystem ;
- User-space applications, with system calls and minimal *libc*

Published articles 1/2

- **Article 1** : Basic PC architecture notions, booting, simple video and Bochs driver ;
- **Article 2** : Interrupt and exception handling. *x86* segmentation ;
- **Article 3** : Physical memory management ;
- **Article 4** : Virtual memory concepts, *x86* paging, implementation of associated routines. *Identity-mapping* and *mirroring* ;
- **Article 5** : Kernel virtual memory allocator ;

Published articles 2/2

- **Article 6** : *Thread* concept. Stack and context switches ;
- **Article 7** : High-level mechanisms for kernel threads : scheduling and synchronization ;
- **Article 8** : *Process* concept, user level threads, separate address spaces, system calls, first userspace applications ;
- **Article 9** : Virtual regions, file and anonymous mappings, copy-on-write, `mmap` and friends, `fork`, `exec` ;
- **Article 10** : Unix-like VFS layer to support filesystems.

Current status

- First article: June 2004, 1200 source code lines ;
- Last article published so far: July 2005, 15700 source code lines ;
- Totalling 136 pages, from 6 to 18 per article ;
- Sos used as a basis for *Toy Lovelace*, an adaptation in *Ada 95*.

Future work 1/2

Three remaining articles :

- Hardware interaction through I/O ports, IRQs, DMA. Simple device drivers : keyboard, serial, IDE ;
- Simple real filesystem driver ;
- Small NE2000 network card driver, and tiny network protocol stack ;

Future work 2/2

Later :

- Produce a compilation of all articles, updated and easier to download ;
- Translate the articles into english, for a broader audience.
- Your contribution ?

Questions ?

Questions ?

Sos :

`http://sos.enix.org`