

# Command line and initramfs in stos

Paul 'Dettorer' Hervot

EPITA 2016

July 19, 2014

Command line  
and initramfs  
in stos

Paul 'Dettorer'  
Hervot

Stos

The command  
line

What it is  
How to use it  
How it works

Initramfs

What and Why  
In stos!

Questions



Stos

# A highly modular monolithic kernel

- Inspired by Linux.
  - Monolithic (everything is in kernel land).
- But much more modular.
  - Only one big core feature: module loading.

Command line  
and initramfs  
in stos

Paul 'Dettorer'  
Hervot

Stos

The command  
line

What it is

How to use it

How it works

Initramfs

What and Why

In stos!

Questions

# The command line

Kernel invocation by the bootloader:

In linux

```
linux /vmlinuz root=/dev/mapper/root ro dolvm
```

root/boot/grub/menu.lst (in stos)

```
kernel /boot/boot.grub module.variable=value
```

# A completely random example

```
root/boot/grub/menu.lst
```

```
kernel /boot/boot.grub initramfs.path=/boot/initrd
```

- Module: `initramfs`
- Argument's name: `path`
- Given value: `/boot/initrd`

# Module dependencies

One of the first loaded modules.

A simple dependency declaration:

```
kernel/modules/initramfs/initramfs.c
```

```
MODINFO {  
    module_name("initramfs"),  
    module_init_once(init),  
    module_deps(M_COMMANDLINE),  
    module_type(M_INITRAMFS)  
};
```

# Declare an argument

Command line  
and initramfs  
in stos

Paul 'Dettorer'  
Hervot

Stos

The command  
line

What it is

How to use it

How it works

Initramfs

What and Why

In stos!

Questions

```
kernel/modules/initramfs/initramfs.c
```

```
char* initramfs_path = NULL;
module_param(path, &initramfs_path);
```

```
include/kernel/module_loader.h
```

```
int add_section_loader(const char* name,
    struct reloc* (*load)(Elf_Ehdr* elf, Elf_Shdr* shdr));

struct reloc* default_load_section(
    Elf_Ehdr* elf, Elf_Shdr* shdr);

int register_pre_init_module_handler(
    int (*handler)(struct module* mod));
```

# Their use by the command line module

Command line  
and initramfs  
in stos

Paul 'Dettorer'  
Hervot

Stos

The command  
line

What it is

How to use it

How it works

Initramfs

What and Why

In stos!

Questions

kernel/modules/commandline/commandline.c

```
static void __init_once init(void)
{
    add_section_loader(".modparam", default_load_section);
    register_pre_init_module_handler(load_params);
}
```

Command line  
and initramfs  
in stos

Paul 'Dettorer'  
Hervot

Stos

The command  
line

What it is

How to use it

How it works

**Initramfs**

What and Why

In stos!

Questions

# Initramfs

# Basic example

Imagine the following partitions:

- /boot: ext2
- /home: ext4 encrypted
- /: ext4 encrypted

Problem: decipher /.

The `initramfs` (or `initramdisk`, `initrd`, `initcpio`):

- Little `cpio` archive.
- A simple file architecture.
- A few binaries.
- A tiny `init` program.

To build under linux: `mkinitcpio!`

## Module loading

`initramfs` module:

- Load the `initramfs` in RAM.

`init` module:

- If `initramfs` available: use it's `init` program.

## Init

`initramfs`' `init`:

- Decipher / then launch system's `init`

An initramfs can be used for a lot of things:

- Root filesystem on LVM, LUKS, remote. . .
- Provide a rescue shell.
- Customize the boot process (a welcome message).
- Third party drivers and modules.
- . . .
- Basically anything possible in userspace that the kernel doesn't do.

```
root/boot/grub/menu.lst
```

```
kernel /boot/boot.grub initramfs.path=/boot/initrd
```

```
initramfs_launch_init:
```

- If path was successfully mounted.
  - Launch initramfs' init (execve).
- Else:
  - Return error

Originally:

- Mount /dev (devfs)
- Mount /dev/hd/1 to /mnt
- Launch /mnt/bin/init

Now:

- Mount /dev (devfs)
- ***Try to run initramfs\_launch\_init***
- ***If initramfs\_launch\_init returned (with error):***
  - Mount /dev/hd/1 to /mnt
  - Launch /mnt/bin/init

Command line  
and initramfs  
in stos

Paul 'Dettorer'  
Hervot

Stos

The command  
line

What it is  
How to use it  
How it works

Initramfs

What and Why  
In stos!

Questions

# Questions

# Thanks!

Questions? Remarks? Worries? Things unclear?

dettorer@lse.epita.fr

*or*

lse@dettorer.net

<http://lse.epita.fr/projects/stos.html>