Operating System Initialization

LISHA/UFSC

Prof. Dr. Antônio Augusto Fröhlich
guto@lisha.ufsc.br
http://www.lisha.ufsc.br/~guto

March 2002
EPOS

- Initialization goals
  - Isolation of architectural dependencies
    - SETUP
    - Hardware mediators
  - Minimum memory overhead
    - Isolate all initialization code to dispose it off latter
      - INIT
  - Mighty first application
    - Possesses all resources
    - Single application in the system or a loader that loads subsequently applications
EPOS

**BOOT I**
- Loads the boot image
- Disables interrupts
- Activates the MMU (if applicable)
- Jumps into SETUP skipping the ELF header
EPOS

- **BOOT II**
  - Relocatable
  - Loads SETUP
  - Allocates SETUP a stack
  - Jumps into SETUP
  - Can usually be merged with BOOT I

Diagram:

```
+---------------------------------+            +---------------------------------+
| setup (code + data)             |  -------->  | setup (stack)                   |
| setup                           |            | setup                           |
| init                            |            | init                            |
| epos                            |            | epos                            |
| app                             |            | app                             |
```
- **SETUP**
  - Initializes hardware components
  - Sets up an initial AS
  - Loads INIT an EPOS
  - Allocates INIT a stack
  - Jumps into INIT
**EPOS**

- **INIT**
  - Initializes EPOS components
  - Loads the application creating the first process
EPOS

- First application
  - Possesses all resources
  - Either
    - The single application
    - A loader that loads subsequent applications

- EPOS architectures
  - built-in
  - μ-kernel
    - system call interface installed
    - OS mapped into all address spaces at the same position in supervisor mode