
Development Guidelines, Commit and Release Cycles

2006/10/01

Stefan Reinauer <stepan@coresystems.de>



core
systems

What are Development Guidelines?

- Rules that all developers agreed on, that handle
 - requirements of the code
 - what the code looks like
 - what the comments look like
 - what the documentation looks like
 - how commits to the repository work
 - how new features are implemented
 - how bug reports are handled
 - project focus and project goals
 - License issues



Why Development Guidelines?

- Helps new Developers to get into LinuxBIOS
- We are always in a hurry :-)
- We all have different interests in LinuxBIOS.
("What's wrong? My mainboard works fine")
- The larger a software project gets, the harder it is to maintain



Requirements

- GNU development environment:
 - GCC
 - binutils
 - python
 - bash
 - IASL
 - ... ?



Coding Guidelines

- Encapsulate and isolate assembly language
- Code shall not be “commented out”
- No use of floating point arithmetics
- No hiding of identifiers defined in outer scopes
- Typedefs are unique (device_t?)
- Functions shall have prototype declarations
- Local functions should be declared static.
- ...



Coding Guidelines II

- No definitions in header files
- All variables are assigned before use.
- All objects should have fully qualified types (unsigned int instead of unsigned)
- Many more of these are described in MISRA-C: 2004 (Guidelines for the use of C in critical systems)



Coding Style

- Indentation: 8
- Kernighan & Ritchie style

```
$ indent -kr -i8 file.c
```

- self-explanatory variable and function names



Comments and Documentation

LinuxBIOS - Mozilla Firefox

http://snapshots.linuxbios.org/docs/doxygen/via_epia-m/

void dev_configure (void)

Configure devices on the devices tree.

Starting at the root of the device tree, travel it recursively in two passes. In the first pass, we compute and allocate resources (ranges) required by each device. In the second pass, the resources ranges are relocated to their final position and stored to the hardware.

I/O resources start at DEVICE_IO_START and grow upward. MEM resources start at DEVICE_MEM_START and grow downward.

Since the assignment is hierarchical we set the values into the dev_root struct.

Definition at line 627 of file `device.c`.

References `dev_root`, `DEVICE_IO_START`, `report_resource_stored()`, and `resource_max()`.

Referenced by `hardwaremain()`.

Here is the call graph for this function:

```
graph LR
    dev_configure --> report_resource_stored
    dev_configure --> resource_max
    report_resource_stored --> dev_path
    report_resource_stored --> resource_type
    report_resource_stored --> resource_end
    resource_max --> align_down
```

Automatic documentation:

<http://snapshots.linuxbios.org/docs/doxygen/>



core
systems

Comments and Documentation

- To create a doxygen comment, write

```
/**  
 * this is a doxygen comment  
 */
```

or

```
/** doxygen comment */
```

- There are a few commands that describe what kind of comment you are adding:
 - @brief – short description of a function
 - @param – input parameters of a function
 - @return – return value of a function
- List of all commands is available at <http://www.stack.nl/~dimitri/doxygen/commands.html>



Comments and Documentation

- Documentation should be put into the WIKI or made available as PDF
- Avoid using different styles and looks of documentation
- There's a documentation directory in the source tree ;-)
- Document „what“, not „how“
- Document assumptions, stipulations etc...
- Document design and concepts!
- Not **lots** of documentation but **good** documentation
- Structured documentation
- Documentation for users, developers, vendors, ...?



Commits to the Repository

- Commits have to be done with a commit comment. This may be short, but descriptive:
 - Code cleanup
 - File/feature xy added
 - Bugfix: broken dram init on k8 (bugzilla #1234)
 - Patch from John Doe <john@doe.gov>
 - Signed off by: Ron Minnich <rminnich@lanl.gov>
 - ...



New Features, Bugs, etc

- There's bugzilla installed at <http://bugzilla.openbios.org/>
- Changes that impact a lot of code **MUST** be documented in bugzilla.
- Please convince another developer to approve such changes before doing a commit.
- Bugs should be documented in bugzilla
- Discussion, communication on the mailing list



Project Focus / Project Goals

- Short term goals?
- Long term goals?
- Milestones? What needs to be done before v3, during v3, ... when would we start v4?
- What does “v3” mean? Is it a release, a release cycle, a spring-cleaning?



License Issues

- License issues need to be documented. Uwe Hermann suggested cleaning up all files, which is a very good idea
- Do we allow binary only payloads?
- Do we allow VGA bios in LinuxBIOS images?
How can we distribute those?



Q&A – Discussion



core
systems