

Adrien Forest
fjxokt@gmail.com
Christophe Carasco
carasco.christophe@gmail.com
Matthieu Maury
mayeu.tik@gmail.com



UPPSALA
UNIVERSITET

*Department of Information Technology
Master Student*

Operating System Project 10hp

SSIK : Simply & Stupidly Implemented Kernel Project Convention

May 19, 2010

Contents

1	Project	3
1.1	Folder hierarchie	3
1.2	Makefile	4
2	Coding convention	5
2.1	Code formating	5
2.2	Comment formating	5
2.2.1	General information	5
2.2.2	File comment	5
2.2.3	Function comment	5
3	Good use of git	7

1. Project

The project name, **SSIK**, standing for: “Simply & Stupidly Implemented Kernel”, is the reversed acronym of **KISS**. **KISS** is an acronym for the design principle “keep it simple and stupid”.

With this project we will try to follow this as a main rule.

1.1 Folder hierarchy

```
root
|- bin/
|- build/
|- doc/
|   |- kernel
|   |- user
|- doc_conf_kernel
|- doc_conf_user
|- include/
|- Makefile
|- README
|- report/
|- scripts/
|- src/
|   |- kernel/
|   |   |- include/
|   |   |- user/
```

Description of all the folder:

bin/: contain the final binarie

build/: contain the object file

doc/kernel/: contain the kernel documentation

doc/user/: contain the user documentation

doc_conf_kernel: doxygene configuration file for the kernel doc

doc_conf_user: doxygene configuration file for the user doc

include/: contain the API and system library header file

Makefile: the project Makefile. Support many option, see further

README: captain obvious prevent me to explain this

report/: contain project report, such as this document

scripts/: script folder, mainly for simisc

src/kernel/: contain the code of the kernel

src/kernel/include: contains general header for the system (like mips related header)

src/user/: contain the code of all the user software

1.2 Makefile

The make file propose this option:

all: build everything

clean: clean every build

doc: generate user and kernel doc

indent: indent all the source with the indent software

kernel: only compile the kernel part. But don't link anything

kerneldoc: generate kernel doc

user: only compile the user part without the system library. But don't link anything

userdoc: generate user doc, which correspond to the user API

run: build everything and run the system in simisc

2. Coding convention

2.1 Code formating

We follow the GNU coding conventions (<http://www.gnu.org/prep/standards/standards.html#Writing-C>) with this modification:

- the block bracket inside a functions should not be indented (`-bli0` in indent)
- no space between a function name and the parenthesis (`-npcs` in indent)
- comment at the right of the code should be aligned (`-c33` in indent)
- name of variable in declaration should be aligned on a column (`-di16` in indent)
- use space instead of space (`-nut` in indent)

The makefile provide an indent option to easily indent everything.

2.2 Comment formating

2.2.1 General information

- comment starting by `/**` will be indexed, not thos started by `/*`
- use at least a brief description with `@brief`. To add a long comment, add a white line after the brief comment
- this tag can be used to comment any structures, variables, etc.

2.2.2 File comment

```
/**
 * @file file name
 * @brief Brief description
 *
 * Long description
 */

/* end of file filename.x */
```

2.2.3 Function comment

```
/**
 * @Brief description which ends at this dot.
 *
 * Details follow here.
 */
```

```
* @param a param
* @return the return value
*/
```

3. Good use of git