TP1 - MiniKernel with Display Management
Advanced System Programming

Rudolf Höhn & Axel Fahy

November 4, 2015

1 Project structure

```
.
├── common
│   ├── base.c
│   ├── base.h
│   ├── colors.h
│   ├── controller_asm.s
│   ├── screen.c
│   ├── screen.h
│   ├── string.c
│   ├── string.h
│   └── types.h
├── doc
│   ├── rapport_fahy_hohn.pdf
│   └── rapport_fahy_hohn.tex
├── grub
│   └── grub.cfg
├── kernel
│   ├── bootloader.s
│   ├── const.inc
│   ├── gdt_asm.s
│   ├── gdt.c
│   ├── gdt.h
│   ├── kernel.c
│   ├── kernel.h
│   ├── kernel.ld
│   ├── makefile
│   └── x86.h
├── makefile
├── test
│   ├── test_cases.c
│   └── test_cases.h
├── tools
└── user
```

base.*
Functions to handle memory.
Implementation of *memset*, *memcpy* and *strncmp*.

controller_ asm.s
Functions to access peripherals and communicate with them.
Implementation of *outb*, *outw*, *inb*, and *inw*.

screen.*
Functions for the screen management.
These files contain functions to initialize the screen and use it. It allows to change background and text color and print string.

We use a structure for the cursor and colors management :

```
typedef struct screen {
    ushort* cursor;
    uchar textColor;
    uchar bgColor;
} screen;
```

The *cursor* attribute contains the current position of the cursor. Each time a character is written, this variable is updated. The *textColor* and *bgColor* attributes are used to know the current color.

```
void setTextColor(uchar color);
```

This function sets the current color of the character. For example, the current font color is white and you set the color using this function, only the next characters will have the new color. All the characters already written keep their font color.

```
void setAllTextColor(uchar color);
```

This function sets the current color of the character and the color of all the other characters already written.

```
void setBackgroundColor(uchar color);
void setAllBackgroundColor(uchar color);
```

The difference between these two functions is the same as the functions which sets the text color but these are for the background of characters.

string.*
Functions to use strings easier.
Implementation of *itoa* which converts an integer to an array of char and *xtoa* which converts an hexadecimal to an array.

colors.h
Definitions of colors.

test_cases.*
Testing file.
We test different things, all of them are listed below :

- **Color** : Update the text color and the background color.
- **Scroll** : Print numbers on the first column of the screen.
- **Scroll2** : Writing a lot of characters.
- **printf** : Call *printf* with one or multiple arguments of different types.
- **Cursor** : Call *setCursorPosition* and *getCursorPosition* and compare the values by printing them on the screen.

doc
The doc folder contains the report.

2 Execution

There are two modes of execution :

- **Test** : Test of functions. To run this mode, run *make* this way :

```
make run MODE=-DTEST
```

- **Default** : Default mode start the kernel and print a welcome message.

```
make run
```

3 Work repartition

For this project, we worked together. We had to think about how we wanted to realize each parts. It would have been more difficult to work separately because of the architecture of the project and its complexity.