

## Lab 4

- Log on **Linux**
- On *System tab*, open **virtualbox**
- *File - Import - Appliance*
- Open Appliance
- On *File System*: `/export/home/corsi/laface/Debian-6.0-rev4-2012.ova`
- Next
- Scroll to the end
- Change *path of the virtual disk image* to
- `/var/tmp/Debian 6.0 laf-2012-disk1.vmdk`
  
- Check/install **nasm**
- Check/install **ddd**
- Check / install the machine emulator and virtualizer **qemu**
  
- Read the file **Booting a PC to run a kernel.pdf**
- Read the file **Installing GRUB on a hard disk image file.pdf**
- Follows with the **ddd** debugger the steps from bootstrap to the **GRUB** kernel selection menu
  
- No report is necessary for this.

### Lab 4.1

Follows the steps for creating the kernel and the bootable hard **hd.img** file in directory

**MK/paging**

Use the tutorial in [http://www.jamesmolloy.co.uk/tutorial\\_html/](http://www.jamesmolloy.co.uk/tutorial_html/) to understand the main and the functions of this minimal kernel.

- Change the file **main.c** to verify which the first address that produces a page fault is.  
Hint: Increment `ptr`, starting from 0, of the page dimension (take care of the pointer arithmetic).  
Use the functions **monitor\_write**, **monitor\_write\_dec**, **monitor\_write\_hex** to print lines such as:  
**Normal access at address 0x0 at page 0**  
**Normal access at address 0x1000 at page 1**  
**.. .. .**

Take note of the **number of pages** that your kernel has allocated.  
Why you get that number?

## Lab 4.2

Write another file **main1.c** that modifies the previous one

Writes the page number, **multiplied by 3**, on the first address of each page

- Prints the content of the content of the first address of all mapped pages (the number information is the one derived from previous main). Ex.:

```
prt 0x0 (page 0) contains 0
prt 0x1000 (page 1) contains 3
prt 0x2000 (page 2) contains 6
.. .. .. .. ..
```

Debug this kernel, modifying **entry 5** of the page table to be not present.

Breakpoint at function **page\_fault** to follow what checks it performs, and what it prints.

In **ddd** command window, you can see the content of variables using command:

- **p var**
- **x/format &var** (or **x/x 0xabcd** for an address)  
where the **format** can be [number] x for hexadecimal, c for characters etc.
- you can select the variable and have it shown in the data window.

You can modify the content of a variable in ddd using

- **p var=newvalue**
- you can select a variable shown in the data window, and modify its value

Produce a report that illustrates the data structure of the directory page table and page table, and the values of a few entries in that tables.

Use figures and tables to make your report easily readable.