STOS - TP1

Gabriel Laskar <gabriel@lse.epita.fr>

Outline

- Clone the repository
- Compile STOS
- How to run STOS in Qemu
- First Module: `hello.ko`
- First Test: `test-hello.ko`

Clone STOS

```
git clone git://git.lse.epita.fr/stos-student
```

Compile STOS

```
$ cd stos-student
$ mkdir build && cd build
$ ../configure --arch=i386 \
               --with-debug \
               --with-klog-serial
$ make bootable
```

Qemu

```
qemu-system-x86_64 \
  -nographic \
  -serial stdio \
  -monitor pty \
  stos-i386-pc.boot
```

First Module: hello.ko

- Create a new directory
- Add module to `$(SUBDIRS)`
- Add config variable `CONFIG_HELLO` to `$BUILD/stos-config`

First Module: hello.ko

```c
#include <kernel/klog.h>
#include <kernel/module.h>
#include <kernel/stos.h>

static void __init_once lol(void)
{
     klog("Init once Hello World !\n");
}

static void __init init(void)
{
     klog("Hello World !\n");
}

MODINFO {
     module_name("hello"),
     module_init(init),
     module_init_once(lol)
};
```

First Test Module: test-hello.ko

```c
#include <kernel/klog.h>
#include <kernel/module.h>
#include <kernel/stos.h>
#include <kernel/test.h>

static void fail(struct gtest *test)
{
        tprint("%s: FAIL\n", test->name);
        panic("this is crap");
}

static void my_test(struct gtest *test)
{
        tprint("%s: %s\n", test->name, "my TEST");
        test->res = 1;
}
```

```c
static void __init init(void)
{
        register_gtest("test-hello", my_test,
                                NULL, fail);
}

MODINFO {
        module_name("test-hello"),
        module_init(init),
        module_deps(M_TEST)
};
```