Compile smallest custom kernel

Configure, build and test your own kernel for the x86 64 bit architecture from Linuses git tree. Smallest as possible.

What to submit:

- Gzipped custom configuration kernel file in format XXXX_config.gz, where **XXXX** should be substituted with your IS MUNI login name
- Exact command line with parameters that was used to test the kernel in qemu
- Achieved size of the **bzImage** with the compiled modules
- Time spent on assignment in hours
- Subject of the mail should have the form
  `KDLP25 E01 VX`
  Where **X** is the number of the revision of the email

Procedure:

Create a Small Test Linux System

!!! This step is not mandatory. Needed only if you want to test the kernel manually without a distribution !!!

Following instructions will help you to create a small linux distribution

- Create a disk image
  `$ dd if=/dev/zero of=minimal.img bs=1024 count=32K`
- Partition and create a primary partition, it using fdisk
  `# fdisk minimal.img`
- Create a filesystem
  `# mkfs.ext4 minimal.img`
- Create a mount point and mount new image as root
  `# mkdir loop`
  `# mount -o loop -t ext4 minimal.img loop`
- Download, compile and install busybox. Make sure you compile a static binary (no shared libraries), you will probably need to install static version of glibc

```
$ wget -v
https://busybox.net/downloads/busybox-1.37.1.tar.bz2
$ tar xjof busybox-1.37.1.tar.bz2
$ cd busybox
$ make menuconfig
$ make
# make CONFIG_PREFIX=../loop/ install
```

- Umount mount point
  `# sync; umount loop`

Compile a kernel

- Clone Linuses git tree.
- Checkout release version `6.16`
- Do "`make defconfig`" to load default configuration (try to not cheat and avoid using tiny config)
- Start the configuration process using "`make menuconfig`" (You will probably have to install several system packages like gcc, make, ncurses-devel)
- Build the kernel using "`make`"
- Repeat the configuration and build steps until you achieve the smallest bootable kernel (including kernel modules)
- Hint, you can copy the modules to a separate directory to check for their size using "`make modules_install INSTALL_MOD_PATH=<path>`"

Requirements:

- The compiled kernel must boot on a default QEMU x86 64-bit machine with ext4 FS and a ttyS0 console.

Points

- Maximum points for this assignment are 10.

Notes

- QEMU - is virtualization tool, that allows you to run operating systems like Linux without creating a special partition on your computer

- Busybox - is an open source project, a set of core Linux utilities in a single executable file that can help you to create a small bootable Linux system.