Building a Kernel

CS 311

We're kernel hackers now. Building a kernel isn't for the faint of heart. **Pay attention to everything below and take copious notes when we discuss this material**, or you **will** find yourself starting over from the beginning with a new virtual machine image. Starting from a shell/terminal:

```
# The following "cd" simply ensures you're in your home directory.

cd

# Let's start off by configuring your global git tooling.  You'll only need
# to run the following commands once, unless your blow-up your VM and have
# to start over, or start working from another account.  Use identity
# values that make sense for you _and_ that others will understand.

git config --global user.name "John Doe"
git config --global user.email JohnDoe@example.com
git config --global push.default simple
git config --global core.editor "emacs -nw" # or just emacs or gedit or vi

# Okay, it can be pretty annoying to have to enter your GitHub credentials
# every single time you interact with your remote, so let's get some
# help.  Run one of these two commands:

#  1) Cache credentials in memory for one hour (units are seconds).  Adjust
#  the timeout as you see fit.

git config --global credential.helper "cache --timeout=3600"

#  2) Store credentials unencrypted on disk permanently
#     Default storage file is ~/.git-credentials

git config --global credential.helper store

# If you're paranoid, to delete the credential cache before the timeout
# expires run

git credential-cache exit

# If you haven't already done so, go into GoucherLearn and follow the link
# you'll find there to create your team and your team's GitHub repository.
# Your team's GitHub repository is <new-repository> below.
# <old-repository> below is the linux-2.6.27.1 repository in GitHub.
# Follow this rule when you create your team name:
#    <initials of 1st member>_<initials of 2nd member>_1_SYSC
# Example:
#    JLZ_TPK_1_SYSC
#
# Okay, now we're ready to duplicate the master Linux kernel repository.
# As discussed, go to the GitHub site to get the URLs for <old-repository>
# and <new-repository>.  One member of your team should run these commands:

git clone --bare <old-repository>.git  # Retrieve the remote master repo

cd <old-repository>.git

git push --mirror <new-repository>.git # Duplicate the master repo
                                         to your personal remote repo

cd ..

# This is an industrial-strength command.  Be careful or you'll wipe out
# far more than you intend.

/bin/rm -rf <old-repository>.git

# Create the real local clone of your remote repo and also create a set of
# working files. Your remote GitHub repository will be configured as the
# remote repository.  Your partner can also use this command to make a
local copy of this repository for collaborative purposes.

git clone <new-repository>.git

# Location for kernel object files.

mkdir build

# cd into your git repository

cd <new-repository>

# It's dot dot slash build slash dot config !!!!

cp minimal.config ../build/.config

# The character preceding the "=" below is the letter O, NOT the
# numeral 0.  This will keep the kernel object files out of the kernel
# source tree.  (Why mix the two?)
#
```

```
# In the menuconfig tool, run below in the next step, select 'General
# setup' and then select 'Local
# version'.  Change the local version string to something like _TPK_00 .
# You can use the two digit number as a version number, incrementing it
# each time you add a new kernel feature.
# Exit the tool, saving the new .config file.

make O=/home/kdev/build menuconfig

make -j 3 O=/home/kdev/build

# The following command will install the kernel modules and the kernel.

sudo make O=/home/kdev/build modules_install install

# Reboot.  Your shiny new kernel should be available as one of the kernel
# choices in GRUB.  Are your brave enough to boot your shiny new kernel?

KABOOM!

# This isn't really necessary, since you can use sudo, but here's how you
# gain the full root environment from the kdev account:

su -

# Exit the root account, returning to the kdev account:

exit

# It can't hurt to change both the kdev and root passwords.  (You never
# know what Tom might do if he breaks into your VM!  1337 h4x0r pwnz j00!)
# Run the following commands as root:

passwd root
passwd kdev
```