Kernel recompilation/optimisation

The kernel provided by all distros is probably the best one for almost everybody. The Linux kernel is compiled in many different ways. Recompiling the kernel can decrease kernel size and its overall behaviour. Changing certain parameters in the source code might also yield some change in the system's performance. Recompile the Linux kernel with different options, install it and then boot the system from it.

Kernel recompilation steps: Download the kernel from the official site, as follows:

```
linux-4.2.1.tar
cd /usr/src/
tar xf /path/to/linux-4.2.1.tar
```

Clear the old object files or configuration information, as follows:

```
make mrproper
```

In the kernel configure menu list, choose which features and drivers you need to compile, as follows:

```
make menu config make clean
make depend
```

After configuring the kernel, make a compressed image of the kernel, as shown below:

```
make bzImage
ls -l arch/x86/boot/bzImage
```

Compile the kernel modules, as follows:

```
make modules
```

Runnable kernels are expected to be in the */boot* directory.

```
cp arch/x86/boot/bzImage /boot/vmlinuz-4.2.1
```

To install kernel modules in the */lib/modules/* directory, use the following command:

```
make modules install
```

Set the *initrd* image of the kernel that provides the initial file systems that get loaded into memory during the Linux startup process, as follows:

```
mkinitrd or initramfs /boot/initrd-4.2.1.img 4.2.1
```

After installing the kernel, the GRUB file has to be modified because it contains the details of the kernels local directory and version:

```
sudo update-grub
```

KGDB

change this line as:

```
linux   /boot/vmlinuz-4.0.0 root=UUID=ebc69508-10b6-4838-95aa-8a47427a1fb6 ro  quiet splash $vt_handoff

linux   /boot/vmlinuz-3.16.1 root=UUID=1d233696-8211-4340-b70b-8595e e532d11 ro kgdboc=ttyS0,115200 kgdbwait
```

Press **F10** to reboot

You would get a black screen stating waiting for remote gdb conncetion

Step 4:

Now run gdb for the vmlinux file that you received from the target machine

i) gdb vmlinux

ii) (gdb) set serial baud 115200

```
           or
      (gdb) set remotebaud 115200
```

iii) (gdb)target remote /dev/ttyS0

iv) Now you can list the code if u set a breakpoint and give run command

```
       eg) (gdb) b set_bits
              (gdb) run
              (gdb) list
```

Step 5: You can safely run the kernel of the target machine by giving continue command

```
                  eg)(gdb)continue (or)c
```

You can the target machine would have started up safely

Hits :

i) both the kernel of target and host machine need **not be same** kernel's

ii) The vmlinux file has to be sent because KGDB symbols are enabled in it and vmlinux is the kernel executable and is unique for each kernel version