LAB - using KGDB

kernel directory: /opt/armsystem/linux-2.6.32
board directory: /opt/armsystem/output
board root file system: /opt/armsystem/output/rootfs

compile the kernel with kgdb support and debugging information
    configure in kernel hacking section
    run make
    copy zImage to the board directory

add kgdb support to the kernel command line
    kgdbwait
    kgdboc=ttyAMA0

add serial device to qemu (before the kernel command line)
    -serial pty

run qemu and look at the host console to see the char device (should be /dev/pts/3)
    ./run_qemu

go to kernel directory and run the debugger
    arm-none-linux-gnueabi-gdb ./vmlinux

(gdb) target remote /dev/pts/3
(gdb) b sys_read
(gdb) c
(gdb) delete breakpoints
(gdb) c

enter debugger from shell
    echo "g">/proc/sysrq-trigger

compile the kernel module in /opt/examples/kernelspce/complexchardrv and copy it to the board root file system

add a kernel module:
1. on target insert module
2. find the section addresses:
    cat /sys/module/[name]/sections/.text
    cat /sys/module/[name]/sections/.data
    cat /sys/module/[name]/sections/.bss

3. gdb
(gdb) add-symbol-file /path/to/module.ko [text address] -s .bss [bss addr] -s .data [data addr]
(gdb) b myread
(gdb) c

use cat on /proc/driver/my_proc_file to enter the code