kgdb注册了reboot notifier，所以当系统reboot时，kgdb也有机会介入。

```c
static void kgdb_register_callbacks(void)
{
    if (!kgdb_io_module_registered) {
        kgdb_io_module_registered = 1;
        kgdb_arch_init();
        if (!dbg_is_early)
            kgdb_arch_late();
        register_module_notifier(&dbg_module_load_nb);
        register_reboot_notifier(&dbg_reboot_notifier);
        atomic_notifier_chain_register(&panic_notifier_list,
                                       &kgdb_panic_event_nb);
#ifdef CONFIG_MAGIC_SYSRQ
        register_sysrq_key('g', &sysrq_dbg_op);
#endif
        if (kgdb_use_con && !kgdb_con_registered) {
            register_console(&kgdbcons);
            kgdb_con_registered = 1;
        }
    }
}
```

```c
static int
dbg_notify_reboot(struct notifier_block *this, unsigned long code, void *x)
{
    /*
     * Take the following action on reboot notify depending on value:
     *    1 == Enter debugger
     *    0 == [the default] detatch debug client
     *   -1 == Do nothing... and use this until the board resets
     */
    switch (kgdbreboot) {
    case 1:
        kgdb_breakpoint();
    case -1:
        goto done;
    }
    if (!dbg_kdb_mode)
        gdbstub_exit(code);
done:
    return NOTIFY_DONE;
}
```

由kgdbreboot variable决定kgdb是否介入。

```c
/* Action for the reboot notifiter, a global allow kdb to change it */
static int kgdbreboot;
```

> module_param(kgdbreboot, int, 0644);

> 所以用户可以通过sysfs来enable / disable该变量（`/sys/module/debug_core/parameters/kgdbreboot`）。

整个流程大致如下：

1. system reboot
2. dbg_notify_reboot() callback run
3. kgdbreboot = 1, kgdb_breakpoint() run

```
/**
 * kgdb_breakpoint - generate breakpoint exception
 *
 * This function will generate a breakpoint exception.  It is used at the
 * beginning of a program to sync up with a debugger and can be used
 * otherwise as a quick means to stop program execution and "break" into
 * the debugger.
 */
noinline void kgdb_breakpoint(void)
{
    atomic_inc(&kgdb_setting_breakpoint);
    wmb(); /* Sync point before breakpoint */
    arch_kgdb_breakpoint();
    wmb(); /* Sync point after breakpoint */
    atomic_dec(&kgdb_setting_breakpoint);
}
EXPORT_SYMBOL_GPL(kgdb_breakpoint);
```

其实就是直接运行一条"undefined instruction"

1. kernel generate "undefined instruction" exception
2. enter kgdb's main entry function - kgdb_handle_exception()
   kgdb被激活