in G2 LSP

CONFIG_DEVTMPFS=y

CONFIG_DEVTMPFS_MOUNT=y

```
config DEVTMPFS
        bool "Maintain a devtmpfs filesystem to mount at /dev"
        help
          This creates a tmpfs/ramfs filesystem instance early at bootup.
          In this filesystem, the kernel driver core maintains device
          nodes with their default names and permissions for all
          registered devices with an assigned major/minor number.
          Userspace can modify the filesystem content as needed, add
          symlinks, and apply needed permissions.
          It provides a fully functional /dev directory, where usually
          udev runs on top, managing permissions and adding meaningful
          symlinks.
          In very limited environments, it may provide a sufficient
          functional /dev without any further help. It also allows simple
          rescue systems, and reliably handles dynamic major/minor numbers.

          Notice: if CONFIG_TMPFS isn't enabled, the simpler ramfs
          file system will be used instead.
```

```
obj-$(CONFIG_DEVTMPFS)  += devtmpfs.o
```

```
config DEVTMPFS_MOUNT
        bool "Automount devtmpfs at /dev, after the kernel mounted the rootfs"
        depends on DEVTMPFS
        help
          This will instruct the kernel to automatically mount the
          devtmpfs filesystem at /dev, directly after the kernel has
          mounted the root filesystem. The behavior can be overridden
          with the commandline parameter: devtmpfs.mount=0|1.
          This option does not affect initramfs based booting, here
          the devtmpfs filesystem always needs to be mounted manually
          after the rootfs is mounted.
          With this option enabled, it allows to bring up a system in
          rescue mode with init=/bin/sh, even when the /dev directory
          on the rootfs is completely empty.
```

在bootcmd上加入

devtmpfs.mount=0就可以disable /dev是tmpfs

in drivers/base/devtmpfs.c

```
#if defined CONFIG_DEVTMPFS_MOUNT
static int mount_dev = 1;
#else
static int mount_dev;
#endif
```

```
static int __init mount_param(char *str)
{
        mount_dev = simple_strtoul(str, NULL, 0);
        return 1;
}
__setup("devtmpfs.mount=", mount_param);
```

```
int devtmpfs_mount(const char *mntdir)
{
        int err;

        if (!mount_dev)
                return 0;

    ......

}
```

这里的mntdir就是在该目录下mount tmpfs

devtmpfs.mount=X(非零)就可以使得/dev为tmpfs

in init/do_mounts.c

```
1.  void __init prepare_namespace(void)
2.  {
3.          int is_floppy;
4.
5.          if (root_delay) {
6.                  printk(KERN_INFO "Waiting %d sec before mounting root device...\n
    ",
7.                         root_delay);
8.                  ssleep(root_delay);
9.          }
10.
11.         /*
12.          * wait for the known devices to complete their probing
13.          *
14.          * Note: this is a potential source of long boot delays.
15.          * For example, it is not atypical to wait 5 seconds here
16.          * for the touchpad of a laptop to initialize.
17.          */
18.         wait_for_device_probe();
19.
20.         md_run_setup();
21.
22.         if (saved_root_name[0]) {
23.                 root_device_name = saved_root_name;
24.                 if (!strncmp(root_device_name, "mtd", 3) ||
25.                     !strncmp(root_device_name, "ubi", 3)) {
26.                         mount_block_root(root_device_name, root_mountflags);
27.                         goto out;
28.                 }
29.                 ROOT_DEV = name_to_dev_t(root_device_name);
30.                 if (strncmp(root_device_name, "/dev/", 5) == 0)
31.                         root_device_name += 5;
32.         }
33.
34.         if (initrd_load())
35.                 goto out;
36.
37.         /* wait for any asynchronous scanning to complete */
38.         if ((ROOT_DEV == 0) && root_wait) {
39.                 printk(KERN_INFO "Waiting for root device %s...\n",
40.                         saved_root_name);
41.                 while (driver_probe_done() != 0 ||
42.                         (ROOT_DEV = name_to_dev_t(saved_root_name)) == 0)
43.                         msleep(100);
44.                 async_synchronize_full();
45.         }
46.
47.         is_floppy = MAJOR(ROOT_DEV) == FLOPPY_MAJOR;
48.
49.         if (is_floppy && rd_doload && rd_load_disk(0))
50.                 ROOT_DEV = Root_RAM0;
51.
52.         mount_root();
```

```
53.  out:
54.          devtmpfs_mount("dev");
55.          sys_mount(".", "/", NULL, MS_MOVE, NULL);
56.          sys_chroot(".");
57.  }
```