in do_mounts.c

mount_block_root("/dev/root", root_mountflags);

/dev/root指向rootfs所在的device，比如"/dev/mmcblk1p2"

```
1.  void __init mount_block_root(char *name, int flags)
2.  {
3.          struct page *page = alloc_page(GFP_KERNEL |
4.                                         __GFP_NOTRACK_FALSE_POSITIVE);
                                                                          ①
5.          char *fs_names = page_address(page);
                                                                  ②
6.          char *p;
7.  #ifdef CONFIG_BLOCK
8.          char b[BDEVNAME_SIZE];
9.  #else
10.         const char *b = name;
11. #endif
12.
13.         get_fs_names(fs_names);

            ③
14. retry:
15.         for (p = fs_names; *p; p += strlen(p)+1) {
                                                          ④
16.                 int err = do_mount_root(name, p, flags, root_mount_data);
                ⑤
17.                 switch (err) {
18.                         case 0:
19.                                 goto out;
20.                         case -EACCES:
21.                                 flags |= MS_RDONLY;
22.                                 goto retry;
23.                         case -EINVAL:
24.                                 continue;
25.                 }
26.                 /*
27.                  * Allow the user to distinguish between failed sys_open
28.                  * and bad superblock on root device.
29.                  * and give them a list of the available devices
30.                  */
31. #ifdef CONFIG_BLOCK
32.                 __bdevname(ROOT_DEV, b);
33. #endif
34.                 printk("VFS: Cannot open root device \"%s\" or %s: error %d\n",
35.                                 root_device_name, b, err);
36.                 printk("Please append a correct \"root=\" boot option; here are t
    he available partitions:\n");
37.
38.                 printk_all_partitions();
39. #ifdef CONFIG_DEBUG_BLOCK_EXT_DEVT
40.                 printk("DEBUG_BLOCK_EXT_DEVT is enabled, you need to specify "
41.                        "explicit textual name for \"root=\" boot option.\n");
42. #endif
43.                 panic("VFS: Unable to mount root fs on %s", b);
44.         }
45.
46.         printk("List of all partitions:\n");
```

```
47.             printk_all_partitions();
48.             printk("No filesystem could mount root, tried: ");
49.             for (p = fs_names; *p; p += strlen(p)+1)
50.                     printk(" %s", p);
51.             printk("\n");
52.     #ifdef CONFIG_BLOCK
53.             __bdevname(ROOT_DEV, b);
54.     #endif
55.             panic("VFS: Unable to mount root fs on %s", b);
56.     out:
57.             put_page(page);
58.     }
```

①

allocate one page space

②

```
1.  #define page_address(page) lowmem_page_address(page)
2.
3.  static __always_inline void *lowmem_page_address(const struct page *page)
4.  {
5.          return __va(PFN_PHYS(page_to_pfn(page)));
6.  }
```

其实就是获得该page的virtual address

③

```
1.  static void __init get_fs_names(char *page)
2.  {
3.          char *s = page;
4.
5.          if (root_fs_names) {
                                              (A)
6.                  strcpy(page, root_fs_names);
7.                  while (*s++) {
8.                          if (s[-1] == ',')
9.                                  s[-1] = '\0';
10.                 }
11.         } else {
12.                 int len = get_filesystem_list(page);
          (B)
13.                 char *p, *next;
14.
15.                 page[len] = '\0';
16.                 for (p = page-1; p; p = next) {
                         (C)
17.                         next = strchr(++p, '\n');
18.                         if (*p++ != '\t')
19.                                 continue;
20.                         while ((*s++ = *p++) != '\n')
21.                                 ;
22.                         s[-1] = '\0';
23.                 }
24.         }
25.         *s = '\0';
26. }
```

(A)

在LSP中root_fs_names为NULL

(B)

get_filesystem_list()返回的是已经register的file system的name.

(C)

把get_filesystem_list()返回的信息处理成类似如下

"tmpfs\0ext2\0ext3\0ext4\0proc\0"

④

对file system name进行枚举。

对"tmpfs\0ext2\0ext3\0ext4\0proc\0"而言，就是p依次为

tmpfs

ext2

ext3

ext4

proc

⑤

这里的root_mount_data如下

in init/do_mounts.c

int root_mountflags = MS_RDONLY | MS_SILENT;

即默认mount rootfs是readonly的。

kernel parameter "ro" and "rw"可以修改该参数。

```
static int __init readonly(char *str)
{
        if (*str)
                return 0;
        root_mountflags |= MS_RDONLY;
        return 1;
}

static int __init readwrite(char *str)
{
        if (*str)
                return 0;
        root_mountflags &= ~MS_RDONLY;
        return 1;
}

__setup("ro", readonly);
__setup("rw", readwrite);
```

in kernel-parameters.txt

ro [KNL] Mount root device read-only on boot

rw [KNL] Mount root device read-write on boot

这里的root_mount_data如下

```
static char * __initdata root_mount_data;
static int __init root_data_setup(char *str)
{
        root_mount_data = str;
        return 1;
}

__setup("rootflags=", root_data_setup);
```

in kernel-parameters.txt

rootflags=	[KNL] Set root filesystem mount option string

即用户可以在boot commandline上通过rootflags来添加mount rootfs的参数。

比如

do_mount_root("/dev/root", "ext3", MS_RDONLY, NULL);

目前G2 LSP的rootfs的file system就是ext3。

由于前面运行过

create_dev("/dev/root", ROOT_DEV);

这里的ROOT_DEV即是rootfs "/dev/mmcblk1p2"的device number。

```
1.  static int __init do_mount_root(char *name, char *fs, int flags, void *data)
2.  {
3.          struct super_block *s;
4.          int err = sys_mount(name, "/root", fs, flags, data);
5.          if (err)
6.                  return err;
7.
8.          sys_chdir("/root");
9.          s = current->fs->pwd.dentry->d_sb;
10.         ROOT_DEV = s->s_dev;
11.         printk(KERN_INFO
12.                "VFS: Mounted root (%s filesystem)%s on device %u:%u.\n",
13.                s->s_type->name,
14.                s->s_flags & MS_RDONLY ?  " readonly" : "",
15.                MAJOR(ROOT_DEV), MINOR(ROOT_DEV));
16.         return 0;
17. }
```

==>

实际上就是调用system call

mount("/dev/root", "/root/", "ext3", MS_RDONLY, NULL);

source "/dev/root" --- 即"/dev/mmcblk1p2"

target "/root/"

file system type = "ext3"

在G2 LSP中ext3 file system driver is builtin driver。