

in fs/Kconfig

```
1.  config TMPFS
2.      bool "Tmpfs virtual memory file system support (former shm fs)"
3.      depends on SHMEM
4.      help
5.          Tmpfs is a file system which keeps all files in virtual memory.
6.
7.          Everything in tmpfs is temporary in the sense that no files will be
8.          created on your hard drive. The files live in memory and swap
9.          space. If you unmount a tmpfs instance, everything stored therein is
10.         lost.
11.
12.         See <file:Documentation/filesystems/tmpfs.txt> for details.
```

code is mm/shmem.c

in mm/shmem.c

```
1.  static struct file_system_type shmem_fs_type = {
2.      .owner          = THIS_MODULE,
3.      .name           = "tmpfs",
4.      .mount          = shmem_mount,
5.      .kill_sb        = kill_litter_super,
6.      .fs_flags       = FS_USERNS_MOUNT,
7.  };
8.
9.  int __init shmem_init(void)
10. {
11.     BUG_ON(register_filesystem(&shmem_fs_type) != 0);
12.
13.     shm_mnt = kern_mount(&shmem_fs_type);
14.     BUG_ON(IS_ERR(shm_mnt));
15.
16.     return 0;
17. }
```

in init/Kconfig

```
1.  config SHMEM
2.      bool "Use full shmem filesystem" if EXPERT
3.      default y
4.      depends on MMU
5.      help
6.          The shmem is an internal filesystem used to manage shared memory.
7.          It is backed by swap and manages resource limits. It is also exported
8.          to userspace as tmpfs if TMPFS is enabled. Disabling this
9.          option replaces shmem and tmpfs with the much simpler ramfs code,
10.         which may be appropriate on small systems without swap.
```

tmpfs是依赖shmem来实现的。

如果CONFIG\_SHMEM没有enable,则shmemfs实际上是由ramfs实现的。

即

CONFIG\_SHMEM = Y

devtmpfs depends on tmpfs depends on shmf

CONFIG\_SHMEM = N

devtmpfs depends on tmpfs depends on ramfs

in mm/shmem.c

```
1.  /*
2.  * tiny-shmem: simple shmemfs and tmpfs using ramfs code
3.  *
4.  * This is intended for small system where the benefits of the full
5.  * shmem code (swap-backed and resource-limited) are outweighed by
6.  * their complexity. On systems without swap this code should be
7.  * effectively equivalent, but much lighter weight.
8.  */
9.
10. static struct file_system_type shmem_fs_type = {
11.     .name           = "tmpfs",
12.     .mount          = ramfs_mount,
13.     .kill_sb        = kill_litter_super,
14.     .fs_flags       = FS_USERNS_MOUNT,
15. };
16.
17. int __init shmem_init(void)
18. {
19.     BUG_ON(register_filesystem(&shmem_fs_type) != 0);
20.
21.     shm_mnt = kern_mount(&shmem_fs_type);
22.     BUG_ON(IS_ERR(shm_mnt));
23.
24.     return 0;
25. }
```