Minix 3.2.1 User Space & Kernel Space

VE482 Project3 Presentation

Group 2
Haoran Jin, Yihua Liu, Yu Xiao, Yuxiang Zhou

UM-SJTU Joint Institute

December 2, 2021

Overview

- Kernel Space
- User Space
  - Drivers
  - Servers
  - User-land libs and exes

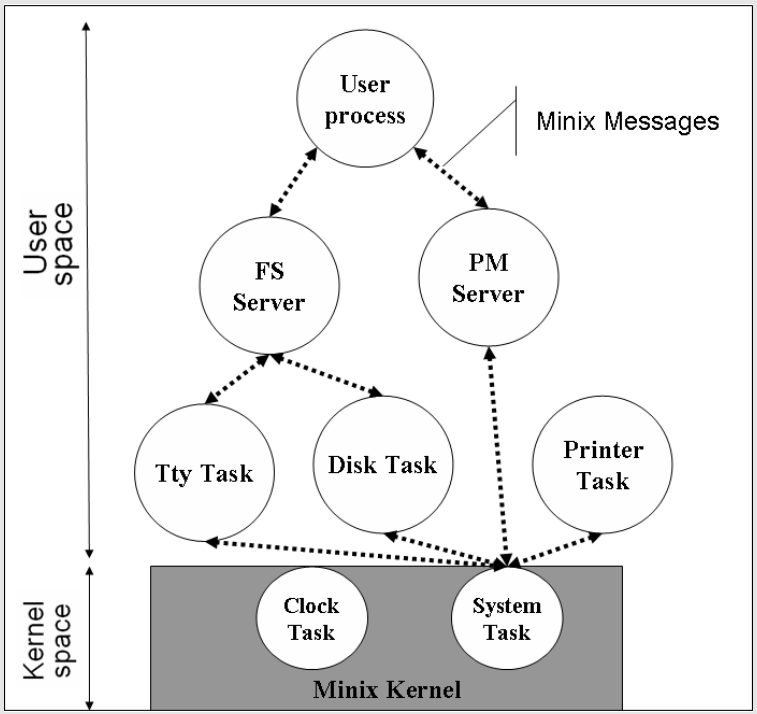

Figure. Overall architecture [9].

Micro-Kernel

Minix 3 is a POSIX-compatible operating system.

- a micro-kernel running a collection of multiple user-mode server processes.
- achieve high reliability
- observing the principle of least authority by limiting what each process has access to and what they can do

Kernel Space

Low-level functionality

- interrupt
- scheduling
- a primitive form of process
- inter-process communication
  - message passing
  - memory grants

Kernel System Call

- Provided by micro-kernel and specific to Minix
- allow the rest system to access message passing, message grants and the hardware
- not POSIX system call(They are implemented at a higher abstraction level)

The Code for Kernel Space

- Kernel source files: `/usr/src/kernel`
- Kernel API: `/usr/src/lib/syslib`

```
leo@leo-Inspiron-7590:~/minix_back/src$ ls kernel
arch       const.h      debug.h            glo.h        kernel.h  proc.c     proto.h    system     type.h            watchdog.c
clock.c    cpulocals.c  extract-errno.sh   interrupt.c  main.c    proc.h     smp.c      system.c   usermapped_data.c watchdog.h
clock.h    cpulocals.h  extract-mfield.sh  interrupt.h  Makefile  profile.c  smp.h      system.h   utility.c
config.h   debug.c      extract-mtype.sh   ipc.h        priv.h    profile.h  spinlock.h table.c     vm.h
```

Figure. Minix kernel files.

Drivers

Drivers are software components used to manage hardware peripherals[5].

- Each driver runs as a separate user-land process and has the controls of some I/O devices.
- Drivers can be categorized into two types:
  - Block Device Drivers
  - Character Device Drivers
- Minix drivers are processes that have permission to access their I/O ports through micro-kernel (kernel calls).
- Minix drivers protocol:
  - The Block Device Protocol
  - The Data Link (inet-ethernet) Protocol
  - The I2C Device Protocol
  - The RTC Protocol

Source Code of Drivers

The source code of drivers: `/usr/src/drivers`

```
# pwd
/usr/src/drivers
# ls
Makefile      at12          fbd           lance         printer       ti1225
Makefile.inc  audio         filter        log           ramdisk       tty
acpi          dec21140A     floppy        memory        random        vbox
ahci          dp8390        fxp           mmc           readclock     virtio_blk
amddev        dpeth         gpio          orinoco       rtl8139       virtio_net
at_wini       e1000         hello         pci           rtl8169
```

Figure. drivers files

Servers

The servers processes servers the other part of the operating system. Different from drivers, servers cannot directly access to hardware resources. Minix servers include the virtual memory, virtual file system, actual file systems, network stack, and data store servers [4]. We will mainly introduce two components: VM (Virtual Memory Manager) and VFS (Virtual File System). The source code of servers are under `/usr/src/servers`.

```
# pwd
/usr/src/servers
# ls
Makefile      ext2          ipc           mfs           rs            vm
Makefile.inc  hgfs          is            pfs           sched
devman        inet          iso9660fs     pm            vbfs
ds            init          lwip          procfs        vfs
```

Figure. servers files

Servers

Virtual Memory Manager

Virtual Memory Manager (VM) manages the usage of the memory in the operating system. There are important data structures in VM describing the memory a process is using in detail[8]:

- Regions: a contiguous range of virtual address space that has a particular type and some parameters described by `struct vir_region` or `region_t`
- Physical regions: a structure exist to reference physical blocks described by `struct phys_region`
- Physical blocks: a single page of physical memory described by `struct phys_block`
- Memory type: abstraction of different behaviour of different memory types described by `struct mem_type`

Servers

Virtual File System

Virtual File System is an abstract layer on top of actual file systems, allowing client applications to access different types of file systems. It provides an interface between the kernel and the file systems, [13]. It can handle most POSIX system calls including:
`access`, `chdir`, `chmod`, `chown`, `chroot`, `close`, `creat`, `fchdir`, `fcntl`, `fstat`, `fstatfs`, `fstatvfs`, `fsync`, `ftruncate`, `getdents`, `ioctl`, `link`, `llseek`, `lseek`, `lstat`, `mkdir`, `mknod`, `mount`, `open`, `pipe`, `read`, `readlink`, `rename`, `rmdir`, `select`, `stat`, `statvfs`, `symlink`, `sync`, `truncate`, `umask`, `umount`, `unlink`, `utime`, `write`.
It also maintains process states shared with PM and VM and keeps track of endpoints that are supposed to be special files of character devices or block devices [7].

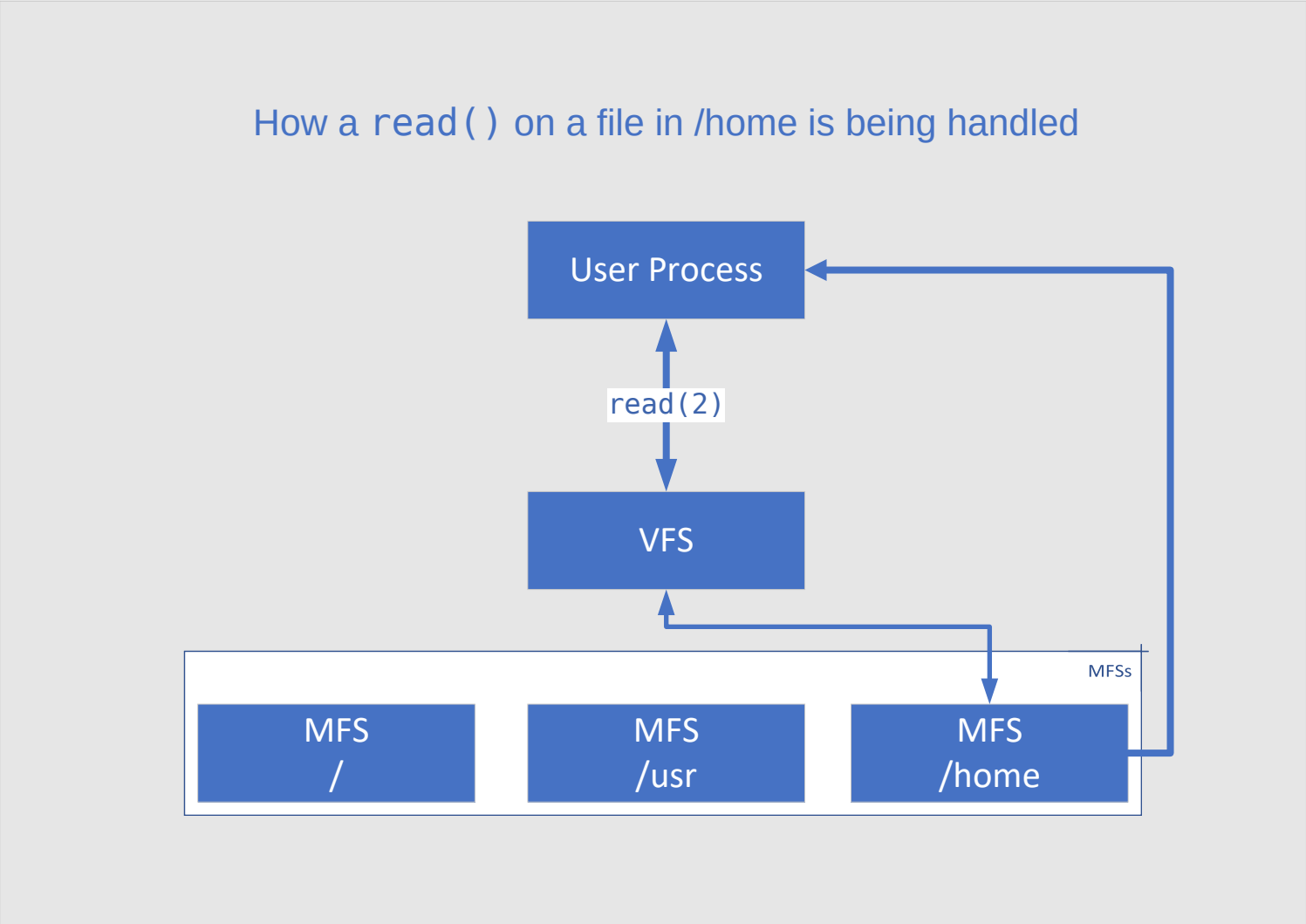

Figure. Handling of `read(2)` system call.

Servers

Other File System Servers

- `pfs`: Pipe File System server. Starting from Minix 3.1.6, pipe handling is removed from file-system drivers to PipeFS.
- `mfs`: Minix File System server. It consists of boot block, superblock, inode bitmap, zone bitmap, inodes area, and data area. It enlightened Linux Torvalds' Extended File System (`ext`).
- `procfs`: Process File System server (starting from Minix 3.2.0). It presents information about processes and other system information as a hierarchical file-like structure widely adopted by Linux, OpenBSD, Solaris, etc [12]. In Minix 3, its internal is a `VTreeFS`, which organization index nodes as a tree by `struct fs_hooks` and `struct inode_stat`. However, security-sensitive information is not dealt with because malicious program may obtain secure values that they should not have permission to [3].

Servers

Other File System Servers

- `ext2`: second extended file system. Support added since version 3.1.8.
- `hgfs` file system. Support added since version 3.1.6. `hgfs` is supported for mounting VMware shared folders as file system.
- `vbfs` file system. Support added since version 3.2.1. `vbfs` is supported for mounting VirtualBox shared folders as file system.
- `iso9660fs`: ISO 9660 file system. It is used to manage optical disc media.

Servers

Other Types of Servers

There are also other types of servers, such as [1]

- `rs`: Reincarnation Server. Reincarnation server is a special feature of Minix 3. [1] It is used to monitor and periodically pings all the drivers and restart them automatically if they crash. It can also kill and restart the looping drivers (making a fresh copy) if they got into an infinite loop and is not responding to status requests [11]. Both problems cannot be solved in monolithic designs.
- `ds`: Data Store Server. It is used to store states and retrieved later. For example, `rs` can make use of it after a crash and subsequent restart [2].

[1] Minix 3 is now dedicated to become a very reliable and safe operating system. Prof. Tennabaum believes that his work is only completed when any users will not get themselves into trouble of system crash [10].

Servers

Other Types of Servers

- `pm`: Process manager.
- `sched`: Scheduler server.
- `is`: Information server. It is used to debug dumps.
- `ipc`: System V IPC server. It works parallel to native minix IPC in kernel.
- `inet` and `lwip`: TCP/IP protocol stack server.
- `init`: Parent of all user processes.
- `devman`: Device manager for hot-plugging of hardware (with `devmand` daemon).

User-land libs and exes

The userland is the collection of libraries and executables that can be used by the users of the operating system, ranging from basic system utilities to desktop environments, web browsers and video games[6].

- Since Minix 3.2.0, Minix has imported huge amounts of userland from the NetBSD project, including bootloader, libc, various utilities and other libraries[11].
- Basically, all the directories in `/usr/src` except for `/usr/src/kernel`, `/usr/src/drivers`, and `/usr/src/servers` belongs to the userland.

User-land libs and exes

You may find system utility `gcc` in `/usr/src/tools`, system utility `cd` in `/usr/src/commands`, and system utility `ls` in `/usr/src/bin`.

```
# pwd
/usr/src
# ls
.git            bin             docs            kernel          servers         usr.sbin
.gitignore      build.sh        drivers         lib             share
.gitreview      commands        etc             libexec         sys
LICENSE         common          external        man             test
Makefile        dist            gnu             releasetools    tools
benchmarks      distrib         include         sbin            usr.bin
```

Figure. directory of userland

Reference I

[1] dcvmoole. *Overview of MINIX3 servers and drivers*. Minix 3. Mar. 22, 2017. URL: https://wiki.minix3.org/doku.php?id=developersguide:overviewofminixservers.

[2] Jorrit N. Herder. *Data Store Server*. Minix 3, Oct. 19, 2005. URL: https://elixir.ortiz.sh/minix/v3.2.1/source/servers/ds/main.c.

[3] Jorrit N. Herder. *David van Moolenbroek*. Minix 3, Feb. 21, 2013. URL: https://elixir.ortiz.sh/minix/v3.2.1/source/servers/procfs/NOTES.

Reference II

[4] jeanbaptisteboric. *Overview of Minix 3 architecture*. Minix 3. Apr. 27, 2015. URL: https://wiki.minix3.org/doku.php?id=developersguide:overviewofminixarchitecture.

[5] jeanbaptisteboric. *Overview of Minix 3 drivers*. Minix 3. Apr. 24, 2015. URL: https://wiki.minix3.org/doku.php?id=developersguide:overviewofminixdrivers.

[6] jeanbaptisteboric. *Overview of Minix 3 userland*. Minix 3. Apr. 26, 2015. URL: https://wiki.minix3.org/doku.php?id=developersguide:overviewofminixuserland.

Reference III

[7] lionelsambuc. *VFS internals*. Minix 3. Nov. 12, 2014. URL: https://wiki.minix3.org/doku.php?id=developersguide:vfsinternals.

[8] lionelsambuc. *VM internals*. Minix 3. Nov. 28, 2014. URL: https://wiki.minix3.org/doku.php?id=developersguide:vminternals.

[9] Pablo Pessolani and Oscar Jara. "Minix over Linux: A User-Space Multiserver Operating System". In: *2011 Brazilian Symposium on Computing System Engineering*. IEEE. 2011, pp. 158–163.

Reference IV

[10] Andrew S. Tanenbaum and Herbert Bos. *Modern Operating Systems*. Chinese. Trans. from the English by Xiangqun Chen and Hongbing Ma. Fourth Edition. Xicheng District, Beijing: China Machine Press, 2017. 610 pp. ISBN: 978-7-111-57369-2.

[11] Wikipedia contributors. *Minix 3 — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-December-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Minix_3&oldid=1055261160.

[12] Wikipedia contributors. *Procfs — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-December-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Procfs&oldid=1056427275.

Reference V

[13] Wikipedia contributors. *Virtual file system — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-December-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Virtual_file_system&oldid=1044838213.

Thanks!