

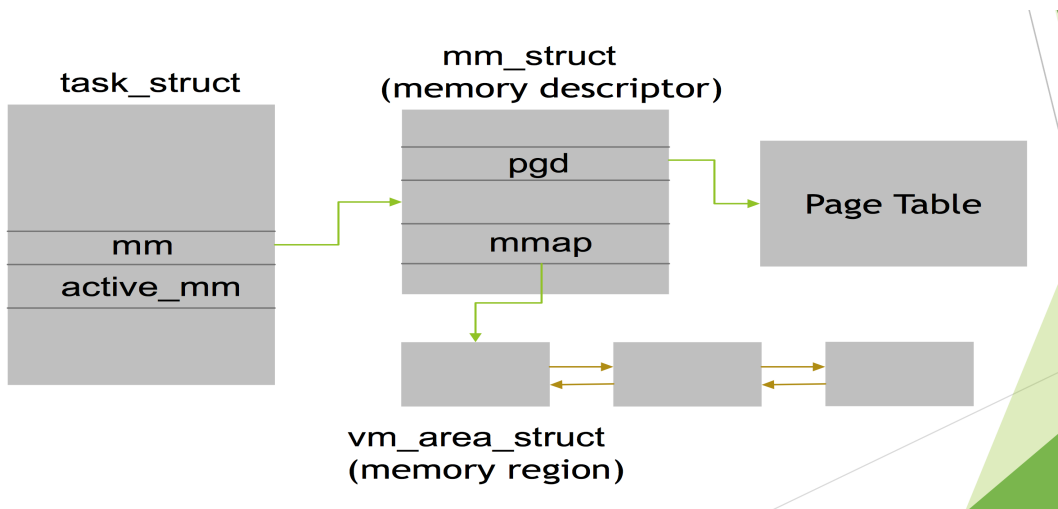
OS Project 3

Team 35

資工三 B03902125 林映廷

資工三 B03902129 陳鵬宇

Trace mmap()



In *linux/sched.h*, *task_struct* of each process contains *mm_struct* in *linux/mm_types.h*. *mm_struct* is memory descriptor, containing informations about the utilization of memory. Its first member is *vm_area_struct*, or memory region, in *linux/mm_types.h*. *vm_area_struct* is a linked list of virtual memory area(VMA).

In addition, for file-backed memory regions, the operation is implemented in *linux/filemap.c*

```
const struct vm_operations_struct generic_file_vm_ops = {
    .fault = filemap_fault,
};
```

Hence, it will invoke *filemap_fault()* when a page fault occurs.

Trace filemap_fault()

When a page fault occurs, the invoked *filemap_fault()* will check whether the required page is in the page cache by *find_get_fault()* at first.

If we find the page in page cache, we will try to call *do_async_mmap_readahead()*; otherwise, we will try to call *do_sync_mmap_readahead()*.

We will find the page by calling *do_async_mmap_readahead()* and *do_sync_mmap_readahead()* if *MADV_RANDOM* isn't in effect. If we find the page by *find_get_page()*, we will lock the page and check whether it is truncated and up-to-date. After checking its size under page lock, we return the required page.

However, if *MADV_RANDOM* is in effect, we *goto no_cached_page* . *no_cached_page* will simply do *page_cache_read()* to read the required page and go back to do *find_get_page()* again.

```
no_cached_page:
/*
 * We're only likely to ever get here if MADV_RANDOM is in
 * effect.
 */
```

Of course, during readahead algorithm, *filemap_fault()* will call *do_async_mmap_readahead()*, *page_cache_async_readahead()*, *ondemand_readahead()*, and *_do_page_cache_readahead()* to page those marked in the page cache for readahead.

Implementation

By tracing *filemap_fault()* , we find two methods to do pure demand paging:

- (1) We return immediately in the beginning of *do_async_mmap_readahead()* and *do_sync_mmap_readahead()* .

```
static void do_async_mmap_readahead(struct vm_area_struct *vma,
                                   struct file_ra_state *ra,
                                   struct file *file,
                                   struct page *page,
                                   pgoff_t offset)
{
    return;
```

```
static void do_sync_mmap_readahead(struct vm_area_struct *vma,
                                   struct file_ra_state *ra,
                                   struct file *file,
                                   pgoff_t offset)
{
    return;
```

- (2) We also comment on the *do_async_mmap_readahead()* and *do_sync_mmap_readahead()* in *filemap_fault()* .

```

page = find_get_page(mapping, offset);
if (likely(page)) {
    /*
     * We found the page, so try async readahead before
     * waiting for the lock.
     */
    //do_async_mmap_readahead(vma, ra, file, page, offset);
    lock_page(page);

    /* Did it get truncated? */
    if (unlikely(page->mapping != mapping)) {
        unlock_page(page);
        put_page(page);
        goto no_cached_page;
    }
} else {
    /* No page in the page cache at all */
    //do_sync_mmap_readahead(vma, ra, file, offset);
    count_vm_event(PGMAJFAULT);
    ret = VM_FAULT_MAJOR;
retry_find:
    page = find_lock_page(mapping, offset);
    if (!page)

```

Pure Demand Paging vs. Readahead Algorithm

(1) Readahead Algorithm

```

edlin@edlin-VirtualBox:~/Desktop/hw3$ sudo ./a.out | tail -3
# of major pagefault: 4200
# of minor pagefault: 2591
# of resident set size: 26664 KB

```

```

edlin@edlin-VirtualBox:~/Desktop/hw3
[ 3.053310] pti4 smbus 0000:00:07:0: SMBus Host Controller at 0x4100, revisi
on 0
[ 3.059464] input: Video Bus as /devices/LNXSYSTM:00/LNXXSYBUS:00/PNP0A03:00/L
NXVIDEO:00/input/input4
[ 3.059496] ACPI: Video Device [GFX0] (multi-head: yes, rom: no, post: no)
[ 3.077433] input: InExP5/2 Generic Explorer Mouse as /devices/platform/l8042
/serio/iinput/input5
[ 3.090683] ACPI: PCI Interrupt Link [LNK0] enabled at IRQ 9
[ 3.090685] PCI: setting IRQ 9 as level-triggered
[ 3.090688] vboxguest 0000:00:04:0: PCI INT A -> Link[LNK0] -> GSI 9 (level,
low) -> IRQ 9
[ 3.091153] vgdvHeartbeatInit: Setting up heartbeat to trigger every 2000 m
iliseconds
[ 3.091243] input: Unspecified device as /devices/pci0000:00/0000:00:04:0/inp
ut/input6
[ 3.092411] vboxguest: misc device minor 57, IRQ 9, I/O port 0020, MMIO at 00
000000f0000000 (size 0x400000)
[ 3.092414] vboxguest: Successfully loaded version 5.1.16 (Interface 0x0001000
04)
[ 3.100247] ppsdev: user-space parallel port driver
[ 3.200044] Bluetooth: Core ver 2.15
[ 3.209052] NET: Registered protocol family 31
[ 3.209053] Bluetooth: HCI device and connection manager initialized
[ 3.209055] Bluetooth: HCI socket layer initialized
[ 3.390653] Intel ICH 0000:00:05:0: PCI INT A -> Link[LNKA] -> GSI 11 (level,
low) -> IRQ 11
[ 3.425000] init: failsafe main process (684) killed by TERM signal
[ 3.529437] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 3.615084] init: alse-restore main process (830) terminated with status 19
[ 3.672971] intel8x0: white list rate for 1028:0177 is 48000
[ 3.821350] vboxsf: Successfully loaded version 5.1.16 (Interface 0x00010004)
[ 4.017504] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: R
[ 4.081805] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 4.109985] VBoxService 5.1.16 r113841 (verbosity: 0) linux.x86 (Mar  8 2017
15:57:03) release log
[ 4.109986] 00:00:00.000175 main    Log opened 2017-06-23T14:14:34.120890000
[ 4.110025] 00:00:00.000241 main    OS Product: Linux
[ 4.110041] 00:00:00.000261 main    OS Release: 2.6.32-60
[ 4.110057] 00:00:00.000277 main    OS Version: #61 SMP Fri Jun 23 21:55:53
CST 2017
[ 4.110078] 00:00:00.000292 main    Executable: /opt/VBoxGuestAdditions-5.1.
16/sbin/VBoxService
[ 4.110078] 00:00:00.000293 main    Process ID: 1172
[ 4.110079] 00:00:00.000294 main    Package type: LINUX_32BITS_GENERIC
[ 4.110113] 00:00:00.001227 main    5.1.16 r113841 started. Verbose level =
[ 4.536536] init: plymouth-stop pre-start process (1304) terminated with stat
us 1
[ 9.811276] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 9.850744] ISO 9660 Extensions: RRIP-1991A
[ 14.516930] eth0: no IPv6 routers present
[ 177.092445] page fault test program starts i
[ 178.902250] page fault test program ends i
edlin@edlin-VirtualBox:~/Desktop/hw3

```

In *Readahead Algorithm*, the pager pre-load more pages. If it guesses correctly on sequential I/O, it will improve the performance; otherwise, it make redundant overhead. Moreover, Its number of major pagefault is more than *Pure Demand Paging*'s one.

(2) Pure Demand Paging

```
edlin@edlin-VirtualBox:~/Desktop/hw3$ sudo ./a.out | tail -3
# of major pagefault: 6566
# of minor pagefault: 224
# of resident set size: 26664 KB
```

```
edlin@edlin-VirtualBox:~/Desktop/hw3
[ 3.223437] vboxguest: misc device minor 57, IRQ 9, I/O port d920, MMIO at 00
000000f0400000 (size 0x400000)
[ 3.223439] vboxguest: Successfully loaded version 5.1.16 (interface 0x000100
04)
[ 3.237105] pti4_smbus 0000:00:07:0: SMBus Host Controller at 0x4100, revisi
on 0
[ 3.314153] ppsdev: user space parallel port driver
[ 3.402502] Bluetooth: core ver 2.15
[ 3.409517] NET: Registered protocol family 31
[ 3.409519] Bluetooth: HCI device and connection manager initialized
[ 3.409520] Bluetooth: HCI socket layer initialized
[ 3.405523] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042
/serio1/input/input4
[ 3.479973] input: Video Bus as /devices/LNXSYSTM:00/LNXSVBUS:00/PNP0A03:00/L
NXVID00:00/input/input5
[ 3.480000] ACPI: Video Device [GFX0] (multi-head: yes rom: no post: no)
[ 3.483880] init: failsafe main process (657) killed by TERM signal
[ 3.523675] usbcore: registered new interface driver hiddev
[ 3.539104] input: VirtualBox USB Tablet as /devices/pcl0000:00/0000:00:06.0/
usb1/1-1/1.1.1.0/input/input6
[ 3.539244] generic-usb 0003:80EE:0021.0001: Input,hidraw: USB HID v1.10 Mou
se [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[ 3.539250] usbcore: registered new interface driver usbhid
[ 3.539250] usbhid: v2.6:USB HID core driver
[ 3.885032] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 3.895891] init:alsa-restore main process (826) terminated with status 19
[ 4.079800] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: R
[ 4.080231] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 4.205105] Intel ICH 0000:00:05:0: PCI INT A -> Link[LINKA] -> GSI 11 (level,
low) -> IRQ 11
[ 4.266909] vboxsf: Successfully loaded version 5.1.16 (Interface 0x00010004)
[ 4.483531] intel8x0: white list rate for 1028:0177 is 48000
[ 4.574421] VBoxService 5.1.16 r113841 (verbosity: 0) linux.x86 (Mar  8 2017
15:57:03) release log
[ 4.574422] 00:00:00.008644 main Log opened 2017-06-23T14:27:53.522240000
[ 4.574506] 00:00:00.009182 main OS Product: Linux
[ 4.574522] 00:00:00.009203 main OS Release: 2.6.32-60
[ 4.574531] 00:00:00.009218 main OS Version: #63 SMP Fri Jun 23 22:24:21
CST 2017
[ 4.574566] 00:00:00.009240 main Executable: /opt/VBoxGuestAdditions-5.1.
16/sbin/VBoxService
[ 4.574567] 00:00:00.009241 main Process ID: 1081
[ 4.574567] 00:00:00.009242 main Package type: LINUX_32BITS_GENERIC
[ 4.577945] 00:00:00.012612 main 5.1.16 r113841 started. Verbose level =
[ 4.838501] init: plymouth-stop pre-start process (1280) terminated with stat
us 1
[ 14.992455] eth0: no IPv6 routers present
[ 157.942787] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 157.968508] ISO 9660 Extensions: RRIP_1991A
[ 222.284757] hrtimer: interrupt took 6354133 ns
[ 500.807206] page fault test program starts !
[ 502.720077] page fault test program ends !
edlin@edlin-VirtualBox:~/Desktop/hw3$
```

In **Pure Demand Paging**, it doesn't pre-load more pages, in contrast to **Readahead Algorithm**. That is, it cannot optimize on sequential I/O. Moreover, Its number of minor pagefault is less than **Readahead Algorithm**'s one.