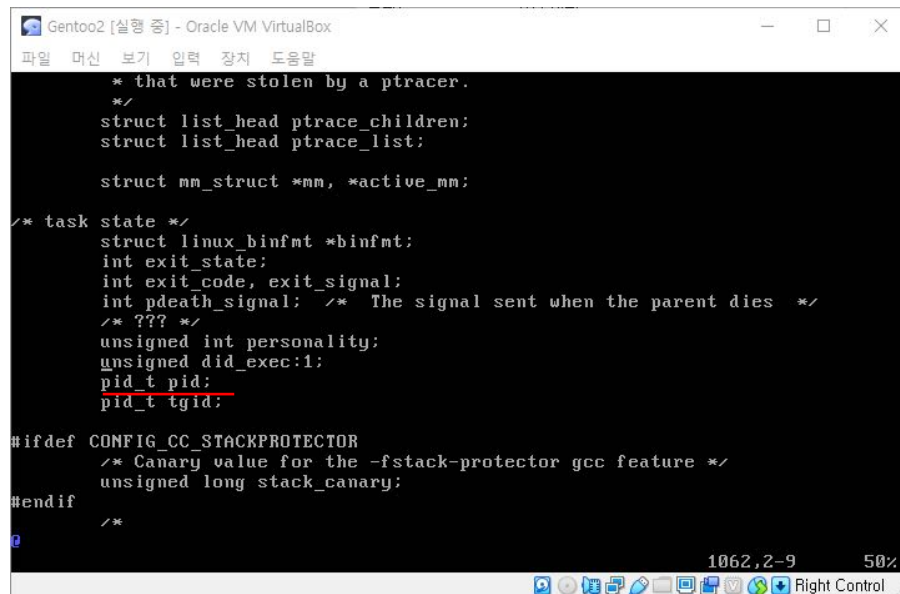


3.1) task\_struct is defined in include/linux/sched.h (search for "task\_struct {"). Which fields of the task\_struct contain information for process id, parent process id, user id, process status, the memory location of the process, the files opened, the priority of the process, program name?



```

/* that were stolen by a ptracer.
 */
struct list_head ptrace_children;
struct list_head ptrace_list;

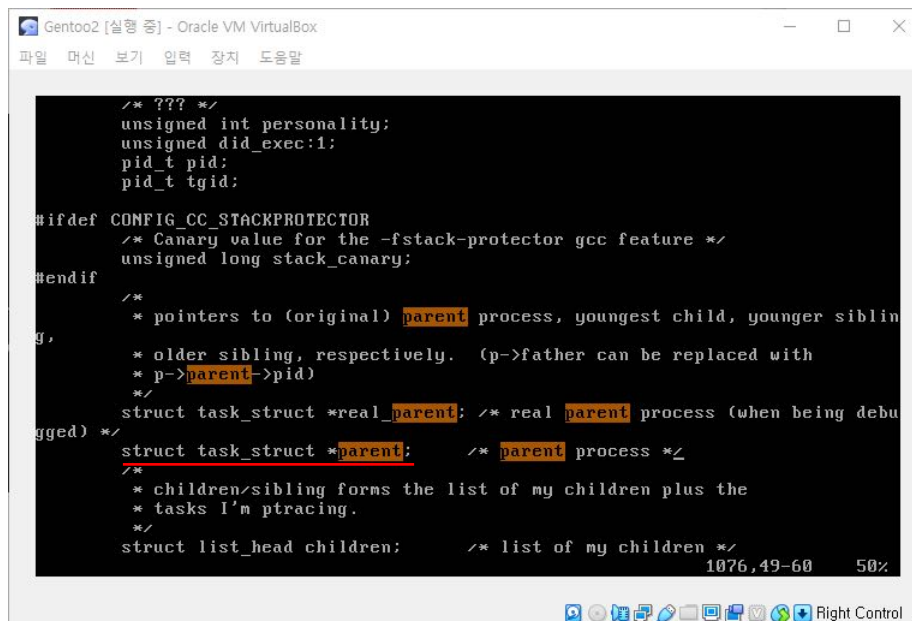
struct mm_struct *mm, *active_mm;

/* task state */
struct linux_binfmt *binfmt;
int exit_state;
int exit_code, exit_signal;
int pdeath_signal; /* The signal sent when the parent dies */
/* ??? */
unsigned int personality;
unsigned did_exec:1;
pid_t pid;
pid_t tgid;

#ifdef CONFIG_CC_STACKPROTECTOR
/* Canary value for the -fstack-protector gcc feature */
unsigned long stack_canary;
#endif
/*

```

<pid>



```

/* ??? */
unsigned int personality;
unsigned did_exec:1;
pid_t pid;
pid_t tgid;

#ifdef CONFIG_CC_STACKPROTECTOR
/* Canary value for the -fstack-protector gcc feature */
unsigned long stack_canary;
#endif
/*
 * pointers to (original) parent process, youngest child, younger sibling,
 * older sibling, respectively. (p->father can be replaced with
 * p->parent->pid)
 */
struct task_struct *real_parent; /* real parent process (when being debugged) */
struct task_struct *parent; /* parent process */
/*
 * children/sibling forms the list of my children plus the
 * tasks I'm ptracing.
 */
struct list_head children; /* list of my children */

```

<ppid>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

    struct timespec start_time;          /* monotonic time */
    struct timespec real_start_time;     /* boot based time */
/* mm fault and swap info: this can arguably be seen as either mm-specific or th
read-specific */
    unsigned long min_flt, maj_flt;

    cputime_t it_prof_expires, it_virt_expires;
    unsigned long long it_sched_expires;
    struct list_head cpu_timers[3];

/* process credentials */
    uid_t uid,euid,suid,fsuid;
    gid_t gid,egid,sgid,fsgid;
    struct group_info *group_info;
    kernel_cap_t cap_effective, cap_inheritable, cap_permitted, cap_bset;
    unsigned keep_capabilities:1;
    struct user_struct *user;
#ifdef CONFIG_KEYS
    struct key *request_key_auth;        /* assumed request_key authority */
    struct key *thread_keyring;          /* keyring private to this thread */
    unsigned char jit_keyring;           /* default keyring to attach requested k
eys to */
#endif
    char comm[TASK_COMM_LEN]; /* executable name excluding path

1119,43-50  52%
```

<user id>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

#ifdef CONFIG_RT_GROUP_SCHED
    struct sched_rt_entity *parent;
    /* rq on which this entity is (to be) queued: */
    struct rt_rq *rt_rq;
    /* rq "owned" by this entity/group: */
    struct rt_rq *my_rq;
#endif
};

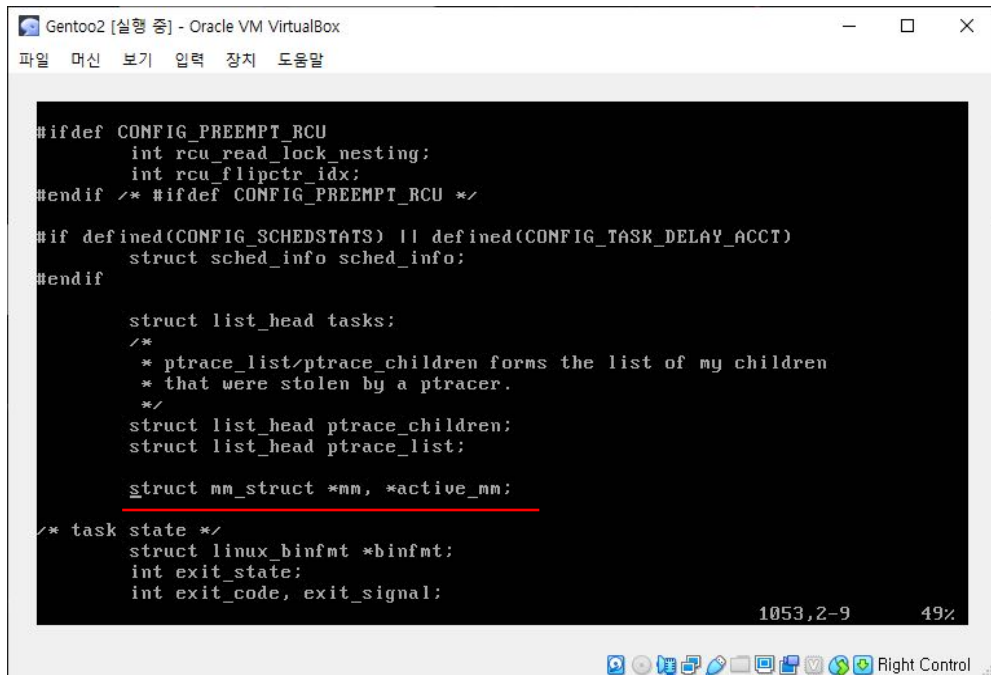
struct task_struct {
    volatile long state; /* -1 unrunnable, 0 runnable, >0 stopped */
    void *stack;
    atomic_t usage;
    unsigned int flags; /* per process flags, defined below */
    unsigned int ptrace;

    int lock_depth; /* BKL lock depth */

#ifdef CONFIG_SMP
#ifdef __ARCH_WANT_UNLOCKED_CTXSW
    int oncpu;
#endif
#endif
}

"sched.h" [converted] 2111L, 63352C  995,2-9  47%
```

<process status>



```
#ifdef CONFIG_PREEMPT_RCU
    int rcu_read_lock_nesting;
    int rcu_flipctr_idx;
#endif /* #ifdef CONFIG_PREEMPT_RCU */

#if defined(CONFIG_SCHEDSTATS) || defined(CONFIG_TASK_DELAY_ACCT)
    struct sched_info sched_info;
#endif

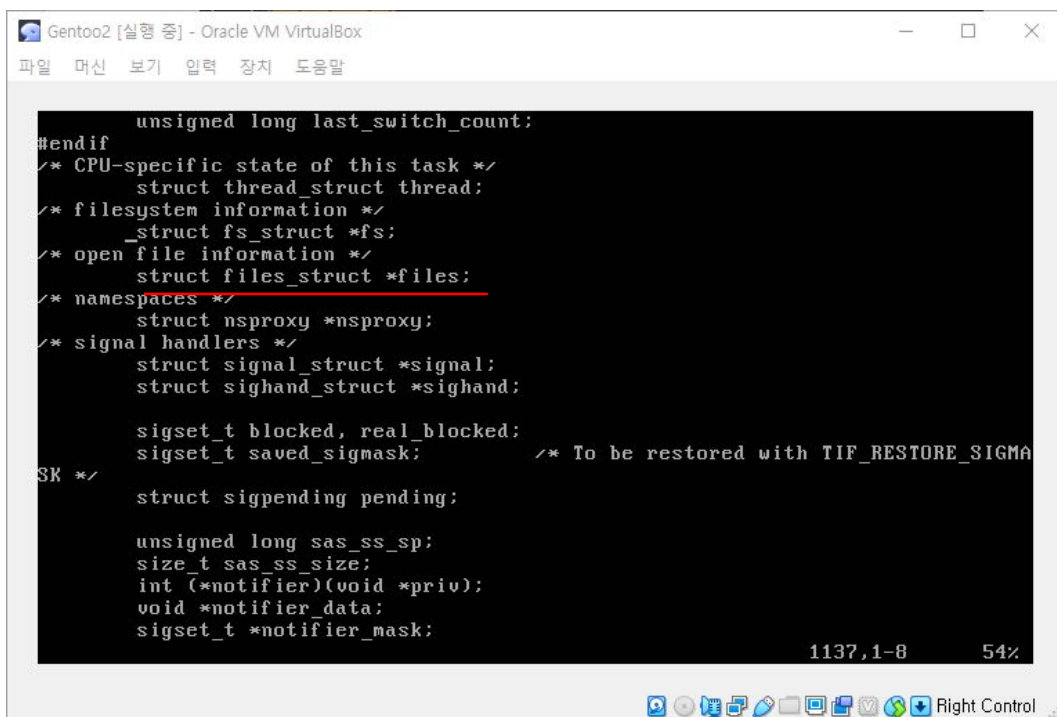
    struct list_head tasks;
    /*
     * ptrace_list/ptrace_children forms the list of my children
     * that were stolen by a tracer.
     */
    struct list_head ptrace_children;
    struct list_head ptrace_list;

    struct mn_struct *mn, *active_mn;

/* task state */
    struct linux_binfmt *binfmt;
    int exit_state;
    int exit_code, exit_signal;
```

1053,2-9 49%

<memory location of process>



```
    unsigned long last_switch_count;
#endif
/* CPU-specific state of this task */
    struct thread_struct thread;
/* filesystem information */
    struct fs_struct *fs;
/* open file information */
    struct files_struct *files;
/* namespaces */
    struct nsproxy *nsproxy;
/* signal handlers */
    struct signal_struct *signal;
    struct sighand_struct *sighand;

    sigset_t blocked, real_blocked;
    sigset_t saved_sigmask; /* To be restored with TIF_RESTORE_SIGMASK */
/*
    struct sigpending pending;

    unsigned long sas_ss_sp;
    size_t sas_ss_size;
    int (*notifier)(void *priv);
    void *notifier_data;
    sigset_t *notifier_mask;
```

1137,1-8 54%

<file opened>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

#endif
};

struct task_struct {
    volatile long state;      /* -1 unrunnable, 0 runnable, >0 stopped */
    void *stack;
    atomic_t usage;
    unsigned int flags;       /* per process flags, defined below */
    unsigned int ptrace;

    int lock_depth;          /* BKL lock depth */

#ifdef CONFIG_SMP
#ifdef __ARCH_WANT_UNLOCKED_CTXSW
    int oncpu;
#endif
#endif

    int prio, static_prio, normal_prio;
    const struct sched_class *sched_class;
    struct sched_entity se;
    struct sched_rt_entity rt;

#ifdef CONFIG_PREEMPT_NOTIFIERS
    struct preempt_notifier *notifier;
#endif
};

991,6  47%
Right Control
```

<priority>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

/* process credentials */
uid_t uid,euid,suid,fsuid;
gid_t gid,egid,sgid,fsgid;
struct group_info *group_info;
kernel_cap_t cap_effective, cap_inheritable, cap_permitted, cap_bset;
unsigned keep_capabilities:1;
struct user_struct *user;

#ifdef CONFIG_KEYS
struct key *request_key_auth; /* assumed request_key authority */
struct key *thread_keyring; /* keyring private to this thread */
unsigned char jit_keyring; /* default keyring to attach requested keys to */
#endif

char comm[TASK_COMM_LEN]; /* executable name excluding path
                          - access with [gs]et_task_comm (which lock
                          it with task_lock())
                          - initialized normally by flush_old_exec */

/* file system info */
int link_count, total_link_count;

#ifdef CONFIG_SYSVIPC
/* ipc stuff */
struct sysv_sem sysvsem;
#endif

1118,6  52%
Right Control
```

<program name>

3.2) Display all processes with "ps -ef". Find the pid of "ps -ef", the process you have just executed. Find the pid and program name of the parent process of it, then the parent of this parent, and so on, until you see the init\_task whose process ID is 0.

```

UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0  0  04:27 ?        00:00:00 init [3]
root      2    0  0  04:27 ?        00:00:00 [kthreadd]
root      3    2  0  04:27 ?        00:00:00 [migration/0]
root      4    2  0  04:27 ?        00:00:00 [ksoftirqd/0]
root      5    2  0  04:27 ?        00:00:00 [watchdog/0]
root      6    2  0  04:27 ?        00:00:00 [events/0]
root      7    2  0  04:27 ?        00:00:00 [khelper]
root     85    2  0  04:27 ?        00:00:00 [kblockd/0]
root     86    2  0  04:27 ?        00:00:00 [kacpid]
root     87    2  0  04:27 ?        00:00:00 [kacpi_notify]
root    130    2  0  04:27 ?        00:00:00 [ata/0]
root    131    2  0  04:27 ?        00:00:00 [ata_aux]
root    132    2  0  04:27 ?        00:00:00 [ksuspend_usbd]
root    137    2  0  04:27 ?        00:00:00 [khubd]
root    140    2  0  04:27 ?        00:00:00 [kseriod]
root    180    2  0  04:27 ?        00:00:00 [pdf_lush]
root    181    2  0  04:27 ?        00:00:00 [pdf_lush]
root    182    2  0  04:27 ?        00:00:00 [kswapd0]
root    225    2  0  04:27 ?        00:00:00 [aio/0]
root    907    2  0  04:27 ?        00:00:00 [scsi_eh_0]
root    924    2  0  04:27 ?        00:00:00 [khpsbpkt]
root    965    2  0  04:27 ?        00:00:00 [kpsmoused]
root    969    2  0  04:27 ?        00:00:00 [kstripped]
  
```

<'ps -ef'를 실행했을 때의 출력문, init[3] 프로세스>

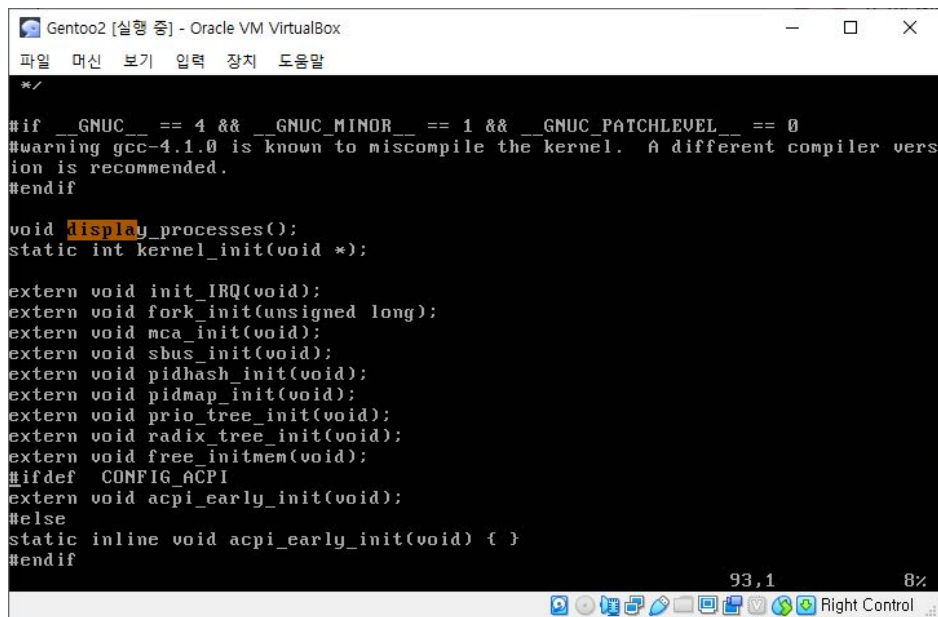
```

root    137    2  0  04:27 ?        00:00:00 [khubd]
root    140    2  0  04:27 ?        00:00:00 [kseriod]
root    180    2  0  04:27 ?        00:00:00 [pdf_lush]
root    181    2  0  04:27 ?        00:00:00 [pdf_lush]
root    182    2  0  04:27 ?        00:00:00 [kswapd0]
root    225    2  0  04:27 ?        00:00:00 [aio/0]
root    907    2  0  04:27 ?        00:00:00 [scsi_eh_0]
root    924    2  0  04:27 ?        00:00:00 [khpsbpkt]
root    965    2  0  04:27 ?        00:00:00 [kpsmoused]
root    969    2  0  04:27 ?        00:00:00 [kstripped]
root    972    2  0  04:27 ?        00:00:00 [kondemand/0]
root    985    2  0  04:27 ?        00:00:00 [rpciod/0]
root   1073    2  0  04:27 ?        00:00:00 [kjournald]
root   1171    1  0  04:27 ?        00:00:00 /sbin/udevd --daemon
root   4344    1  0  04:27 ?        00:00:00 /usr/sbin/syslog-ng
root   4459    1  0  04:27 ?        00:00:00 /usr/sbin/cron
root   4523    1  0  04:27 tty1      00:00:00 /bin/login --
root   4525    1  0  04:27 tty2      00:00:00 /sbin/agetty 38400 tty2 linux
root   4527    1  0  04:27 tty3      00:00:00 /sbin/agetty 38400 tty3 linux
root   4529    1  0  04:27 tty4      00:00:00 /sbin/agetty 38400 tty4 linux
root   4531    1  0  04:27 tty5      00:00:00 /sbin/agetty 38400 tty5 linux
root   4533    1  0  04:27 tty6      00:00:00 /sbin/agetty 38400 tty6 linux
root   4546   4523  0  04:28 tty1      00:00:00 -bash
root   4560   4546  0  04:29 tty1      00:00:00 ps -ef
  
```

<ps -ef, -bash, /bin/login -- 프로세스>

ps -ef의 pid는 4560이고, ppid는 4546이다. -bash의 pid는 4546이고, ppid는 4523이다.  
/bin/login --의 pid는 4523이고 ppid는 1이다. init[3]의 pid는 1이다.

3.3) Define `display_processes()` in `init/main.c` (right before the first function definition). Call this function in the beginning of `start_kernel()`. Confirm that there is only one process in the beginning. Find the location where the number of processes becomes 2. Find the location where the number of processes is the greatest. Use "dmesg" to see the result of `display_processes()`.



```
*/
#if __GNUC__ == 4 && __GNUC_MINOR__ == 1 && __GNUC_PATCHLEVEL__ == 0
#warning gcc-4.1.0 is known to miscompile the kernel. A different compiler vers
ion is recommended.
#endif

void display_processes();
static int kernel_init(void *);

extern void init_IRQ(void);
extern void fork_init(unsigned long);
extern void mca_init(void);
extern void sbus_init(void);
extern void pidhash_init(void);
extern void pidmap_init(void);
extern void prio_tree_init(void);
extern void radix_tree_init(void);
extern void free_initmem(void);
#ifdef CONFIG_ACPI
extern void acpi_early_init(void);
#else
static inline void acpi_early_init(void) { }
#endif
```

<display\_processes()함수 프로토타입 선언>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

    cpu_set(cpu, cpu_present_map);
    cpu_set(cpu, cpu_possible_map);
}

void __init __attribute__((weak)) smp_setup_processor_id(void)
{
}

void display_processes(){
    struct task_struct *temp;
    temp = &init_task;
    for(;;){
        printk("pid : %d, name : %s, state : %d\n", temp->pid, temp->com
            m, temp->state);
        temp = next_task(temp);
        if(temp == &init_task){
            break;
        }
    }
}

-- INSERT --                                     515,9-23      58%
```

<display\_processes()함수 선언>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

        break;
    }
}

asmlinkage void __init start_kernel(void)
{
    display_processes();
    printk("hello from My Linux2\n");
    char * command_line;
    extern struct kernel_param __start___param[], __stop___param[];

    smp_setup_processor_id();

    /*
     * Need to run as early as possible, to initialize the
     * lockdep hash:
     */
    unwind_init();
    lockdep_init();
    cgroup_init_early();

    local_irq_disable();
    early_boot_irqs_off();

-- INSERT --                                     520,2      60%
```

<start\_kernel()함수 내 display\_processes()함수 호출>



```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

kernel_thread(kernel_init, NULL, CLONE_FS | CLONE_SIGHAND);
numa_default_policy();
pid = kernel_thread(kthreadd, NULL, CLONE_FS | CLONE_FILES);
kthreadd_task = find_task_by_pid(pid);
unlock_kernel();

/*
 * The boot idle thread must execute schedule()
 * at least once to get things moving:
 */
init_idle_bootup_task(current);
preempt_enable_no_resched();
schedule();
preempt_disable();

display_processes();

/* Call into cpu_idle with preempt disabled */
cpu_idle();
}

/* Check for early params. */
static int __init do_early_param(char *param, char *val)
453,0-1 50%
```

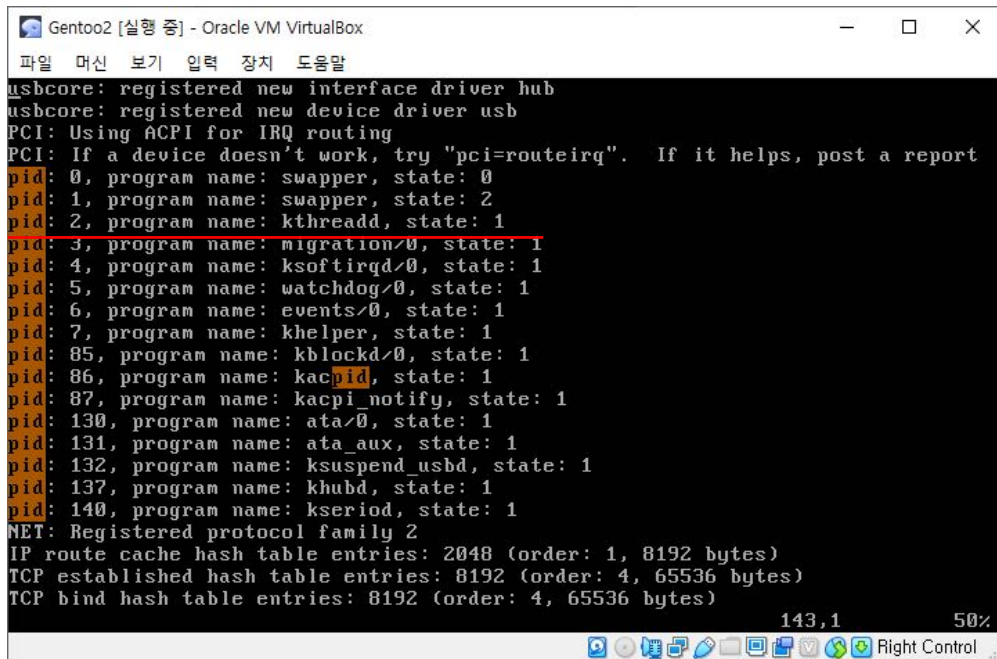
<rest\_init()함수 내 display\_processes()함수 호출, start\_kernel()의 마지막에 rest\_init()함수가 호출되기 때문에, rest\_init()함수 내 display\_processes()함수를 호출하였음>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

pid: 0, program name: swapper, state: 0
hello from my Linux
Linux version 2.6.25.10 (root@localhost) (gcc version 4.1.2 (Gentoo 4.1.2 p1.0.2)) #8 SMP Mon Oct 7 07:49:39 KST 2019
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000a0000 - 0000000000010000 (reserved)
BIOS-e820: 0000000000010000 - 00000000000fff0000 (usable)
BIOS-e820: 00000000000fff0000 - 00000000000100000000 (ACPI data)
BIOS-e820: 00000000000fec0000 - 00000000000fec010000 (reserved)
BIOS-e820: 00000000000fec010000 - 00000000000fec010000 (reserved)
BIOS-e820: 00000000000fec010000 - 00000000000fec010000 (reserved)
BIOS-e820: 00000000000fec010000 - 00000000000fec010000 (reserved)
WARNING: strange, CPU MTRRs all blank?
-----[ cut here ]-----
WARNING: at arch/x86/kernel/cpu/mtrr/main.c:696 mtrr_trim_uncached_memory+0x16f/0x17a()
Modules linked in:
Pid: 0, comm: swapper Not tainted 2.6.25.10 #8
[<c011d667>] warn_on_slowpath+0x40/0x4f
[<c011de05>] release_console_sem+0x180/0x199
[<c011de05>] release_console_sem+0x180/0x199
[<c011de05>] release_console_sem+0x180/0x199
[<c011e24d>] printk+0x14/0x18
"z" [readonly][converted] 307L, 14302C
1,1 Top
```

<dmesg 첫 부분>





```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
PCI: Using ACPI for IRQ routing
PCI: If a device doesn't work, try "pci=routeirq".  If it helps, post a report
pid: 0, program name: swapper, state: 0
pid: 1, program name: swapper, state: 2
pid: 2, program name: kthreadd, state: 1
pid: 3, program name: migration/0, state: 1
pid: 4, program name: ksoftirqd/0, state: 1
pid: 5, program name: watchdog/0, state: 1
pid: 6, program name: events/0, state: 1
pid: 7, program name: khelper, state: 1
pid: 85, program name: kblockd/0, state: 1
pid: 86, program name: kacpid, state: 1
pid: 87, program name: kacpi_notify, state: 1
pid: 130, program name: ata/0, state: 1
pid: 131, program name: ata_aux, state: 1
pid: 132, program name: ksuspend_usbd, state: 1
pid: 137, program name: khubd, state: 1
pid: 140, program name: kseriod, state: 1
NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 4, 65536 bytes)
143,1 50%
Right Control
```

<dmesg 뒷부분>

'kthreadd'라는 실행파일이 특정 프로세스를 실행시켰을 때, pid가 2가 됨

3.4) Make a system call that, when called, displays all processes in the system. Run an application program that calls this system call and see if this program displays all processes in the system.

```

.long sys_oldumount
.long sys_setuid16
.long sys_getuid16
.long sys_stime      /* 25 */
.long sys_ptrace
.long sys_alarm
.long sys_fstat
.long sys_pause
.long sys_utime      /* 30 */
.long sys_start_print /* sys start print */
.long sys_stop_print  /* sys stop print */
.long sys_access
.long sys_nice
.long sys_display_processes /* 35 - old ftime syscall holder */
.long sys_sync
.long sys_kill
.long sys_rename
.long sys_mkdir
.long sys_rmdir      /* 40 */
.long sys_dup
.long sys_pipe
.long sys_times
.long sys_ni_syscall /* old prof syscall holder */
.long sys_brk        /* 45 */

```

37,27-34 7%

<syscall table에 display\_processes명 선언>

```

/*
 * This is one of the first .c files built. Error out early if we have compiler
 * trouble.
 */

#if __GNUC__ == 4 && __GNUC_MINOR__ == 1 && __GNUC_PATCHLEVEL__ == 0
#warning gcc-4.1.0 is known to miscompile the kernel. A different compiler vers
ion is recommended.
#endif

asmlinkage void display_processes();
static int kernel_init(void *);

extern void init_IRQ(void);
extern void fork_init(unsigned long);
extern void mca_init(void);
extern void sbus_init(void);
extern void pidhash_init(void);
extern void pidmap_init(void);
extern void prio_tree_init(void);
extern void radix_tree_init(void);
extern void free_initmem(void);
search hit BOTTOM, continuing at TOP

```

81,17 8%

<main.c파일에 display\_processes() 프로토타입 선언>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

asmlinkage void display_processes(){
    struct task_struct *temp;
    temp = &init_task;
    for(;;){
        printk("pid : %d, program name : %s, state : %d\n", temp->pid,
temp->comm, temp->state);

        temp = next_task(temp);
        if(temp == &init_task){
            break;
        }
    }
}

asmlinkage void __init start_kernel(void)
{
    printk("choi calls this one\n");
    char * command_line;
"main.c" [converted] 878L, 21193C written
localhost init # cd ..
localhost linux-2.6.25.10 #
```

<display\_processes()함수 선언>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

#include<stdio.h>

int main()
{
    syscall(35);
    return 0;
}

"displaysyscalltest.c" [converted] 7L, 59C
7,1 All
Right Control
```

<display\_processes()를 syscall(35)로 호출한 파일>

```
Gentoo2 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말
pid : 138, program name : khubd, state : 1
pid : 141, program name : kseriod, state : 1
pid : 182, program name : pdflush, state : 1
pid : 183, program name : pdflush, state : 1
pid : 184, program name : kswapd0, state : 1
pid : 227, program name : aio/0, state : 1
pid : 911, program name : scsi_ch_0, state : 1
pid : 928, program name : khpsbpkt, state : 1
pid : 969, program name : kpsmoused, state : 1
pid : 973, program name : kstripped, state : 1
pid : 976, program name : kondemand/0, state : 1
pid : 989, program name : rpciod/0, state : 1
pid : 1077, program name : kjournald, state : 1
pid : 1174, program name : udevd, state : 1
pid : 4318, program name : syslog-ng, state : 0
pid : 4433, program name : cron, state : 1
pid : 4497, program name : login, state : 1
pid : 4499, program name : agetty, state : 1
pid : 4501, program name : agetty, state : 1
pid : 4503, program name : agetty, state : 1
pid : 4505, program name : agetty, state : 1
pid : 4507, program name : agetty, state : 1
pid : 4516, program name : bash, state : 0
pid : 4540, program name : display, state : 0
localhost ~ #
```

<해당 c파일을 컴파일한 결과>

3.5) Run three user programs, f1, f2, and f3, and run another program that calls the above system call as follows. State 0 means runnable and 1 means blocked. Observe the state changes in f1, f2, f3 and explain what these changes mean.

```
f1:
    int i,j; double x=1.2;
for(i=0;i<100;i++){
    for(j=0;j<1000000;j++){ // make f1 busy for a while
        x=x*x;
    }
    // and then sleep 1sec
    usleep(1000000);
}

f2:
    int i,j; double x=1.2;
for(i=0;i<100;i++){
    for(j=0;j<1000000;j++){ // make f2 busy for a while
        x=x*x;
    }
    // and then sleep 2sec
    usleep(2000000);
}

f3:
    int i,j; double x=1.2;
for(i=0;i<100;i++){
    for(j=0;j<1000000;j++){ // make f3 busy for a while
        x=x*x;
    }
    // and then sleep 3sec
    usleep(3000000);
}

ex1.c:
    for(i=0;i<100;i++){
        sleep(5);
        syscall(17); // show all processes
                    // assuming the system call number in homework (3.4) is 17
    }

#./f1 &
#./f2 &
#./f3 &
```

#./ex1

```
pid : 183, program name : pdflush, state : 1
pid : 184, program name : kswapd0, state : 1
pid : 227, program name : aio/0, state : 1
pid : 911, program name : scsi_eh_0, state : 1
pid : 928, program name : khpsbpkt, state : 1
pid : 969, program name : kpsmoused, state : 1
pid : 973, program name : kstripped, state : 1
pid : 976, program name : kondemand/0, state : 1
pid : 989, program name : rpciod/0, state : 1
pid : 1077, program name : kjournald, state : 1
pid : 1175, program name : udevd, state : 1
pid : 4350, program name : syslog-ng, state : 0
pid : 4465, program name : cron, state : 1
pid : 4529, program name : login, state : 1
pid : 4531, program name : agetty, state : 1
pid : 4533, program name : agetty, state : 1
pid : 4535, program name : agetty, state : 1
pid : 4537, program name : agetty, state : 1
pid : 4539, program name : agetty, state : 1
pid : 4548, program name : bash, state : 1
pid : 4645, program name : f1, state : 1
pid : 4646, program name : f2, state : 1
pid : 4647, program name : f3, state : 1
pid : 4649, program name : ex1, state : 0
```

<첫 번째 result>

```
pid : 183, program name : pdflush, state : 1
pid : 184, program name : kswapd0, state : 1
pid : 227, program name : aio/0, state : 1
pid : 911, program name : scsi_eh_0, state : 1
pid : 928, program name : khpsbpkt, state : 1
pid : 969, program name : kpsmoused, state : 1
pid : 973, program name : kstripped, state : 1
pid : 976, program name : kondemand/0, state : 1
pid : 989, program name : rpciod/0, state : 1
pid : 1077, program name : kjournald, state : 1
pid : 1175, program name : udevd, state : 1
pid : 4350, program name : syslog-ng, state : 0
pid : 4465, program name : cron, state : 1
pid : 4529, program name : login, state : 1
pid : 4531, program name : agetty, state : 1
pid : 4533, program name : agetty, state : 1
pid : 4535, program name : agetty, state : 1
pid : 4537, program name : agetty, state : 1
pid : 4539, program name : agetty, state : 1
pid : 4548, program name : bash, state : 1
pid : 4645, program name : f1, state : 0
pid : 4646, program name : f2, state : 1
pid : 4647, program name : f3, state : 1
pid : 4649, program name : ex1, state : 0
```

<두 번째 result>

```
pid : 182, program name : pdflush, state : 1
pid : 183, program name : pdflush, state : 1
pid : 184, program name : kswapd0, state : 1
pid : 227, program name : aio/0, state : 1
pid : 911, program name : scsi_eh_0, state : 1
pid : 928, program name : khpsbpkt, state : 1
pid : 969, program name : kpsmoused, state : 1
pid : 973, program name : kstripped, state : 1
pid : 976, program name : kondemand/0, state : 1
pid : 989, program name : rpciod/0, state : 1
pid : 1077, program name : kjournald, state : 1
pid : 1175, program name : udevd, state : 1
pid : 4350, program name : syslog-ng, state : 0
pid : 4465, program name : cron, state : 1
pid : 4529, program name : login, state : 1
pid : 4531, program name : agetty, state : 1
pid : 4533, program name : agetty, state : 1
pid : 4535, program name : agetty, state : 1
pid : 4537, program name : agetty, state : 1
pid : 4539, program name : agetty, state : 1
pid : 4548, program name : bash, state : 1
pid : 4646, program name : f2, state : 1
pid : 4647, program name : f3, state : 0
pid : 4649, program name : ex1, state : 0
```

<세 번째 result>

f1은 1초마다, f2는 2초마다, f3은 3초마다 실행되고 ex1은 5초마다 실행된다. 그에 따라 f1, f2, f3의 state가 계속 바뀐다. state가 0인 것은 프로세스가 실행되고 있다는 것을 의미하고, state가 1인 것은 해당 프로세스가 sleep상태가 되었다는 것을 의미한다.