

# HW3

## Q1

1. Interrupts are changes in the flow of the system generated by the hardware. The interrupt handler is called to handle the cause of the interrupt; then, control will return to the context and instructions of the interrupt. A trap is an interrupt generated by software. Interrupts can be used to signal I/O completion, thereby avoiding device polling. Traps can be used to call operating system routines or catch arithmetic errors
2. The operating system of such machines needs to be in control (or monitoring mode) at all times. This can be done in two ways: a. Software interpretation of all user programs (e.g. some BASIC, Java, and LISP systems). The software interpreter will provide content that is not provided by the hardware in the software. b. All programs are required to be written in a high-level language so that all target code is compiler generated. The compiler generates (inline or through function calls) a protection check for hardware loss.
3. f c a d e g b
4. The processor can track the location associated with each process and restrict access to locations outside the scope of the program. Information about the program memory range can be maintained by using basic registers and limit registers and performing checks on each memory access.

## Q2

we can use the command "cd arch/arm64/kernel" and "vim sys.c" get the code

The screenshot shows a terminal window with the following content:

```
root@fb5cf0adcf: /have-fun-debugging/sys2lab-21fall/src/lab3/linux-5.15.2/arch/arm64/ke...
// SPDX-License-Identifier: GPL-2.0-only
/*
 * ARM64-specific system calls implementation
 * Copyright (c) 2012 ARM Ltd.
 * Author: Catalin Marinas <catalin.marinas@arm.com>
 */
#include <linux/compiler.h>
#include <linux/errno.h>
#include <linux/fs.h>
#include <linux/mm.h>
#include <linux/export.h>
#include <linux/sched.h>
#include <linux/slab.h>
#include <linux/syscalls.h>

#include <asm/cpufeature.h>
#include <asm/syscall.h>

SYSCALL_DEFINE(mmap, unsigned long, addr, unsigned long, len,
               unsigned long, prot, unsigned long, flags,
               unsigned long, fd, unsigned long, off)
{
    if (offset_in_page(off) != 0)
        return -EINVAL;
    return ksys_mmap_pgoff(addr, len, prot, flags, fd, off >> PAGE_SHIFT);
}

SYSCALL_DEFINE1(arm64_personality, unsigned int, personality)
{
    if (personality(personality) == PER_LINUX32 &&
        !system_supports_32bit_el0())
        return -EINVAL;
    return ksys_personality(personality);
}

asmlinkage long sys_nt_syscall(void);
asmlinkage long __arm64_sys_nt_syscall(const struct pt_regs * __unused)
{
    return sys_nt_syscall();
}

/*
 * Wrappers to pass the pt_regs argument.
 */
#define __arm64_sys_personality __arm64_sys_arm64_personality
#define __SYSCALL
#define __SYSCALL(nr, sym) asmlinkage long __arm64_##sym(const struct pt_regs *);
#include <asm/unistd.h>

"sys.c" 61L, 1497C
```

The web browser window shows a homework page with the following content:

homework-3

作业要求 我的作业

占成绩比例	0.0%	公布成绩时间	马上公布
开放时间	2021.11.09 14:48	作业交付截止	2021.11.21 23:59
作业形式	个人作业	完成指标	提交作业

评分方式 (教师评分 100.0%)

教师评分

占成绩比例	100.0%
1. Operating System Concepts Chapter 1 Exercises: 1.14, 1.17, 1.19, 1.22 (20 points)	
2. Detail your steps about how to get arch/arm64/kernel/sys.i (10 points)	
3. Find system call table of Linux v5.15 for ARM32, RISC-V(32 bit), RISC-V(64 bit), x86_64 (50 points)	
List source code file, the whole system call table with macro expanded, screenshot every step.	
4. Explain what is ELF file? Try readelf and objdump command on an ELF file, give screenshot of the output.	
Run an ELF file and cat /proc/PID/maps to give its memory layout.	

2, 3, 4 need to have screenshots.

2004-2020 浙江大学 浙ICP备05074421号 电话: 0571-87951669 / E-mail: xwmaster@zju.edu.cn / 地址: 中

## Q3

we can get the syscall code on Linux for x86 & x86\_64 from arch/x86/entry



we can get the syscalls code on Linux for arm from arch/arm/tools

```
Activities Terminal 11月 20 14:01 root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic

/* fs/loprto.c */
#define __NR_loprto_set 30
__SYSCALL(__NR_loprto_set, sys_loprto_set)
#define __NR_loprto_get 31
__SYSCALL(__NR_loprto_get, sys_loprto_get)

/* fs/locks.c */
#define __NR_flock 32
__SYSCALL(__NR_flock, sys_flock)

/* fs/namel.c */
#define __NR_mknodat 33
__SYSCALL(__NR_mknodat, sys_mknodat)
#define __NR_mkdirat 34
__SYSCALL(__NR_mkdirat, sys_mkdirat)
#define __NR_unlinkat 35
__SYSCALL(__NR_unlinkat, sys_unlinkat)
#define __NR_symlinkat 36
__SYSCALL(__NR_symlinkat, sys_symlinkat)
#define __NR_linkat 37
__SYSCALL(__NR_linkat, sys_linkat)
/* #def __ARCH_WANT_RENAMEAT
/* renameat is superseded with flags by renameat2 */
#define __NR_renameat 38
__SYSCALL(__NR_renameat, sys_renameat)
/* #def __ARCH_WANT_RENAMEAT */

/* fs/namespace.c */
#define __NR_umount2 39
__SYSCALL(__NR_umount2, sys_umount)
#define __NR_mount 40
__SYSCALL(__NR_mount, sys_mount)
#define __NR_pivot_root 41
__SYSCALL(__NR_pivot_root, sys_pivot_root)

/* fs/nfsctl.c */
#define __NR_nfsdsvctl 42
__SYSCALL(__NR_nfsdsvctl, sys_nt_syscall)

/* fs/open.c */
#define __NR264_stats 43
__SC_COMP_3264(__NR3264_stats, sys_stats64, sys_stats, \
__COMPAT_sys_stats64)
#define __NR3264_fstats 44
__SC_COMP_3264(__NR3264_fstats, sys_fstats64, sys_fstats, \
__COMPAT_sys_fstats64)
#define __NR3264_truncate 45
__SC_COMP_3264(__NR3264_truncate, sys_truncate64, sys_truncate, \
__COMPAT_sys_truncate64)
#define __NR3264_ftruncate 46
__SC_COMP_3264(__NR3264_ftruncate, sys_ftruncate64, sys_ftruncate, \
__COMPAT_sys_ftruncate64)
#define __NR_fallocate 47
```

```
Activities Terminal 11月 20 14:01 root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic

root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/riscv/kernel# cd linux-5.15.2
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/riscv/kernel# ls
Makefile crash_dump.c kexec_relocate.S process.o soc.o
asm-offsets.c crash_save_regs.S kexec.c ptrace.o stacktrace.c
asm-offsets.s elf-header.S machine_kexec.c ptrace.o stacktrace.o
built-in.a elf.c mcount-dyn.S reset.c sys_riscv.c
cachelf.o elf.o mcount.S reset.o sys_riscv.o
cachelf.o entry.S module-sections.c riscv_kyms.c syscall_table.c
cpu-hotplug.c entry.o module-sections.o riscv_kyms.o syscall_table.o
cpu-hotplug.o fpu.S module.c sbt.c time.c
cpu.c fpu.o module.o sbt.o time.o
cpu.o ftrace.c modules.order setup.c traps.c
cpu_ops.c head.S patch.c setup.o traps.o
cpu_ops.o head.h patch.c signal.c traps_misaligned.c
cpu_ops_sbi.c head.o perf_calchain.c signal.o vds.o
cpu_ops_sbi.o image-vars.h perf_event.c snp.c vds.o
cpu_ops_splwait.c irq.c perf_regs.c snp.o vds.o
cpu_ops_splwait.o irq.o perf_regs.o snpboot.c vmlinux-xip.lds.S
cpufeature.c jump_label.o probev snpboot.c vmlinux.lds
cpufeature.o jump_label.o process.c soc.c vmlinux.lds.S
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/riscv/kernel# cd linux-5.15.2
bash: cd: linux-5.15.2: No such file or directory
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/riscv/kernel# cd
root@fb5c3f0adcf: # exit
exit
#have-fun-debugging-CSS6-Stall0: $ docker exec -lt oslab1 bash
root@fb5c3f0adcf: /# cd have-fun-debugging
root@fb5c3f0adcf: /have-fun-debugging# cd sys2lab-2ifall/src/lab3
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3# cd linux-5.15.2
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2# ls
COPYING Makefile crypto lib samples vmlinux
CREDITS Module.symvers drivers mm scripts vmlinux.o
Documentation README fs modules-only.symvers security vmlinux.symvers
kbuild System.map include modules.builtin sound
kconfig arch link modules.builtin.modinfo tools
LICENSE blob ipc modules.order usr
MAINTAINERS certs kernel net virt
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2# cd include
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include# ls
api asm-generic crypto generated kvm media net rdma sound uapi xen
linux.h linux.h.tmpl linux.h.tmpl.h linux.h.tmpl.h.tmpl linux.h.tmpl.h.tmpl.h
linux.h.tmpl.h.tmpl.h.tmpl.h linux.h.tmpl.h.tmpl.h.tmpl.h
kbuild asm-generic dma linux mtd rdma test vmx vmx.o
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi# cd asm-generic
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic# ls
kbuild fcntl.h ipcbuf.h poll.h siginfo.h stats.h unistd.h
auxvec.h hugetlb_encode.h kvm_para.h posix_types.h signal-defs.h swab.h
bitsperlong.h int-l64.h mman-common.h resource.h signal.h termios.h
bpf_perf_event.h int-l64.h mman.h sembuf.h socket.h termios.h
errno-base.h ioctl.h nsgbuf.h setup.h sockets.h types.h
errno.h ioctl.h param.h sembuf.h stat.h ucontext.h
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic# cd unistd.h
root@fb5c3f0adcf: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic#
```

```
Activities Terminal 11月20 14:03
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm/tools

Linux system call numbers and entry vectors
#
# The format is:
# <num> <abi> <name> <entry point> <abi compat entry point>]
#
# Where abi is:
# common - For system calls shared between oabi and eabi (may have compat)
# oabi - For oabi-only system calls (may have compat)
# eabi - For eabi-only system calls
#
# For each syscall number, "common" is mutually exclusive with oabi and eabi
#
# common restart_syscall sys_restart_syscall
# common exit sys_exit
# common fork sys_fork
# common read sys_read
# common write sys_write
# common open sys_open
# common close sys_close
# 7 was sys_waitpid
# common creat sys_creat
# common link sys_link
# common unlink sys_unlink
# common execve sys_execve
# common chdir sys_chdir
# oabi ttime sys_ttime32
# common mknod sys_mknod
# common chmod sys_chmod
# common lchown sys_lchown16
# 17 was sys_break
# 18 was sys_stat
# common lseek sys_lseek
# common getpid sys_getpid
# common mount sys_mount
# oabi umount sys_oldumount
# common setuid sys_setuid16
# common getuid sys_getuid16
# oabi stime sys_stime32
# common ptrace sys_ptrace
# oabi alarm sys_alarm
# 28 was sys_fstat
# common pause sys_pause
# oabi utime sys_utime32
# 31 was sys_stty
# 32 was sys_gtty
# common access sys_access
# common nice sys_nice
# 35 was sys_ftime
# common sync sys_sync
# common kill sys_kill
# common rename sys_rename
# common mkdir sys_mkdir
# common rmdir sys_rmdir
```

```
Activities Terminal 11月20 14:03
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm/tools

kbuild System.map include modules.builtin sound
kconfig arch init modules.builtin.modinfo tools
tcwasm block ipc modules.order usr
MAINTAINERS
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2# cd /include
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include# ls
asm config dt-bindings kunit math-emu misc ras soc trace video
asm-generic crypto generated km media net dma sound uapi xen
blacksource drin keys linux memory pci nci sct target vds
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include# cd uapi
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi# ls
asm-generic drin linux misc nci dma sct sound video xen
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi# cd asm-generic
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic# ls
kbuild fcntl.h ipcbuf.h poll.h signal.h statfs.h unistd.h
auxvec.h hugetlb_encode.h kvm_para.h posix_types.h signal-defs.h swab.h
bitops/long.h int-ls.h mman-common.h resource.h signal.h termios.h
bpf/perf_event.h int-ls.h mman.h sembuf.h socket.h termios.h
errno-base.h ioctl.h msgbuf.h setup.h sockios.h types.h
errno.h ioctl.h param.h shmbuf.h stat.h ucontext.h
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic# vln unistd.h
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/include/uapi/asm-generic# cd
root@fb5c3f0adcf4: ~# exit
exit
zhangyc@zhangyc-G566-Stealth:~$ docker exec -it oslab1 bash
root@fb5c3f0adcf4: /# cd /have-fun-debugging
root@fb5c3f0adcf4: /have-fun-debugging# cd sys2lab-2ifall/src/lab3
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3# cd /linux-5.15.2
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2# cd arch
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch# ls
kconfig src smd h8xx lsm microblaze mips2 openrisc powerpc riscv sh um xtensa
alpha arm csky hexagon m8k mips ntos2 parisc riscv sh um xtensa
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch# cd arm
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm# ls
kbuild mach-sa9260 mach-hisi mach-mx mach-sa9260 net
kconfig mach-aspeed mach-lnx mach-homdtk mach-sa1300 nfp
kconfig-nemu mach-atis mach-integrator mach-ngen mach-shmobile plat-onip
kconfig-asmblr mach-axxia mach-lp32x mach-nspire mach-socfpga plat-orion
kconfig-debug mach-bcn mach-lp4xx mach-onapi mach-spear plat-pxa
makefile mach-bellu mach-kgstone mach-onap2 mach-stl plat-versatile
mach-cips71xx mach-lp32xx mach-orionx mach-stb32 probe
common mach-cns3xxx mach-lp32xx mach-oxnas mach-sunxi tools
configs mach-davinci mach-mediatek mach-pxa mach-tegra vds
crypto mach-eflcolor mach-meson mach-qcom mach-omap4 vfp
include mach-dove mach-mtlbeaut mach-rda mach-ux500 xen
kernel mach-ep93xx mach-mmp mach-realtek mach-versatile
lib mach-exynos mach-mstar mach-realview mach-vexpress
mach-actions mach-footbridge mach-mstar mach-rockchip mach-v8500
mach-alpine mach-gemini mach-mv78xx0 mach-rpc mach-zynq
mach-mpu
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm# cd tools
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm/tools# ls
makefile gen-mach-types mach-types syscall.tbl syscallnr.sh
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm/tools# vln syscall.tbl
root@fb5c3f0adcf4: /have-fun-debugging/sys2lab-2ifall/src/lab3/linux-5.15.2/arch/arm/tools#
```

## Q4

The ELF (Executable and Linkable Format), for example the exe/.so/.out/.o/core are ELF. Each ELF is include : 1. Program header table 2. Section header table 3. Data referred to by entries is the program header table or section header table

