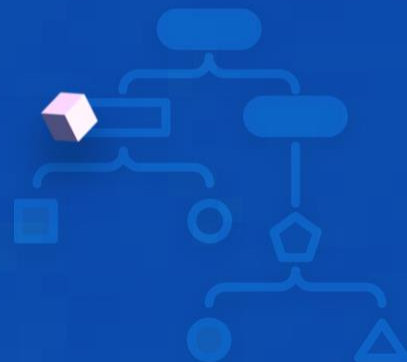


강원혁신플랫폼

리눅스프로그래밍

새로운 시스템 호출 함수 등록





사용자 영역에서 직접 시스템 호출을 할 수 있도록 Linux 커널에서 제공하는 함수로, 시스템 호출 번호로 식별되는 특정 시스템 호출을 요청하는 데 사용되는 함수는 무엇인가요?

`syscall()`






학습 내용

1 새로운 시스템 호출 함수 등록 방법

학습 목표

 새로운 시스템 호출 함수 등록 방법을 설명할 수 있다.

강원혁신플랫폼

리눅스프로그래밍



새로운 시스템 호출 함수 등록 방법



```
arch/arm/include/generated/uapi/asm/unistd-common.h
```

```
#ifndef _UAPI_ASM_ARM_UNISTD_H
```

```
#define _UAPI_ASM_ARM_UNISTD_H 1
```

```
#define __NR_restart_syscall (__NR_SYSCALL_BASE + 0)
```

... 중간 생략

```
#define __fspick (__NR_SYSCALL_BASE + 433)
```

```
#define __NR_pidfd_open_free (__NR_SYSCALL_BASE + 434)
```

```
#define __NR_clone3 (__NR_SYSCALL_BASE + 435)
#define __NR_helloworld (__NR_SYSCALL_BASE + 436)

#endif /* _UAPI_ASM_ARM_UNISTD_H */
```



```
./arch/arm/tools/syscall.tbl
```

```
0    common restart_syscall    sys_restart_syscall
```

```
1    common exit               sys_exit
```

```
2    common fork              sys_fork
```

```
3    common read              sys_read
```

```
4    common write             sys_write
```

... 중간 생략

```
433  common fspick            sys_fspick
```

```
434  common pidfd_open        sys_pidfd_open
```

```
435  common clone3            sys_clone3
```

```
436  common helloworld        sys_helloworld
```



```
include/linux/syscalls.h
```

```
#ifndef _LINUX_SYSCALLS_H
```

```
#define _LINUX_SYSCALLS_H
```

... 중간 생략

```
asmlinkage long sys_helloworld (void);
```

```
#endif
```




```
kernel/helloworld.c
```

```
#include <linux/kernel.h>
```

```
asmlinkage long sys_helloworld(void) // 어셈블러 소스 코드(.s)와 C언어 함수의 링크
```

```
{
```

```
    printk(KERN_INFO "Hello World! from Linux Kernel\n");
```

```
    return 0;
```

```
}
```



새로운 커널 함수 구현

printk() 함수

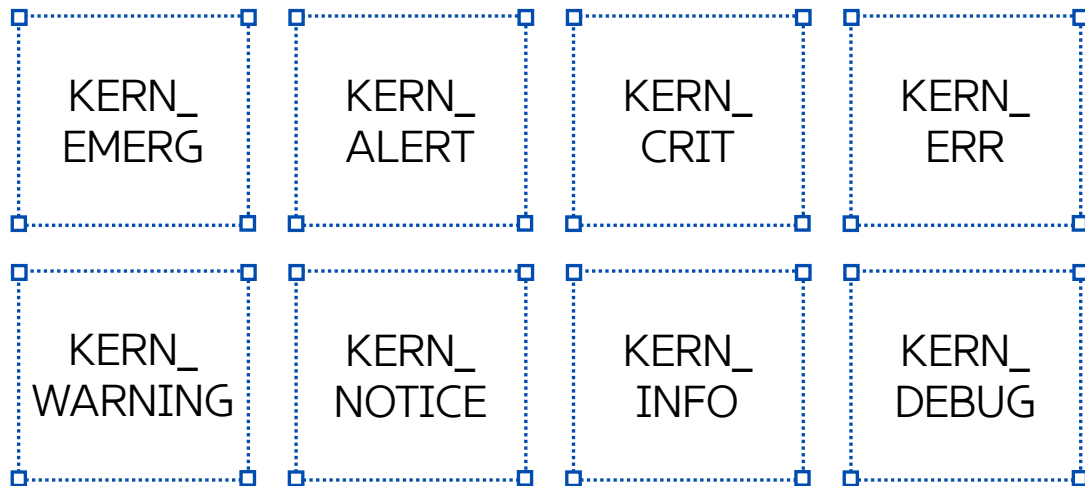
[커널 로그에 메시지를 인쇄할 수 있도록 하는
Linux 커널의 로깅 함수]



새로운 커널 함수 구현



로깅 수준





```
#  
# Makefile for the linux kernel.  
#  
  
obj-y    = fork.o exec_domain.o panic.o \  
          cpu.o exit.o itimer.o time.o softirq.o resource.o \  
/* - 중간 표시 생략 - */  
          notifier.o ksysfs.o cred.o reboot.o \  
          async.o range.o smpboot.o ucount.o helloworld.o  
/* - 중간 표시 생략 - */
```



```
$ sudo make bzImage
```

```
$ sudo cp ./arch/arm/boot/zImage /boot/kernel.img
```

```
$ sudo reboot
```



```
#include <unistd.h>
```

```
#include <linux/unistd.h>
```

```
int helloworld()
```

```
{
```

```
    return syscall(__NR_helloworld);
```

```
}
```

```
int main(int argc, char** argv)
```

```
{
```





```
int ret = helloworld();
```

```
    return ret;
```

```
}
```





```
# makefile
```

```
.SUFFIXES : .c .o
```

```
OBJECTS = helloworld
```

```
SRCS = $(OBJECTS:.o=.c)
```

```
CC = gcc
```

```
CFLAGS = -g
```

```
LDFLAGS =
```

```
TARGET = $(OBJECTS:.o=)
```





```
all : helloworld
```

```
$(TARGET) : $(TARGET:=.c)
```

```
$(CC) -o $@ $@.c
```

```
clean :
```

```
$(RM) $(OBJECTS) $(TARGET) core
```



01 • 새로운 시스템 호출 함수 등록 방법