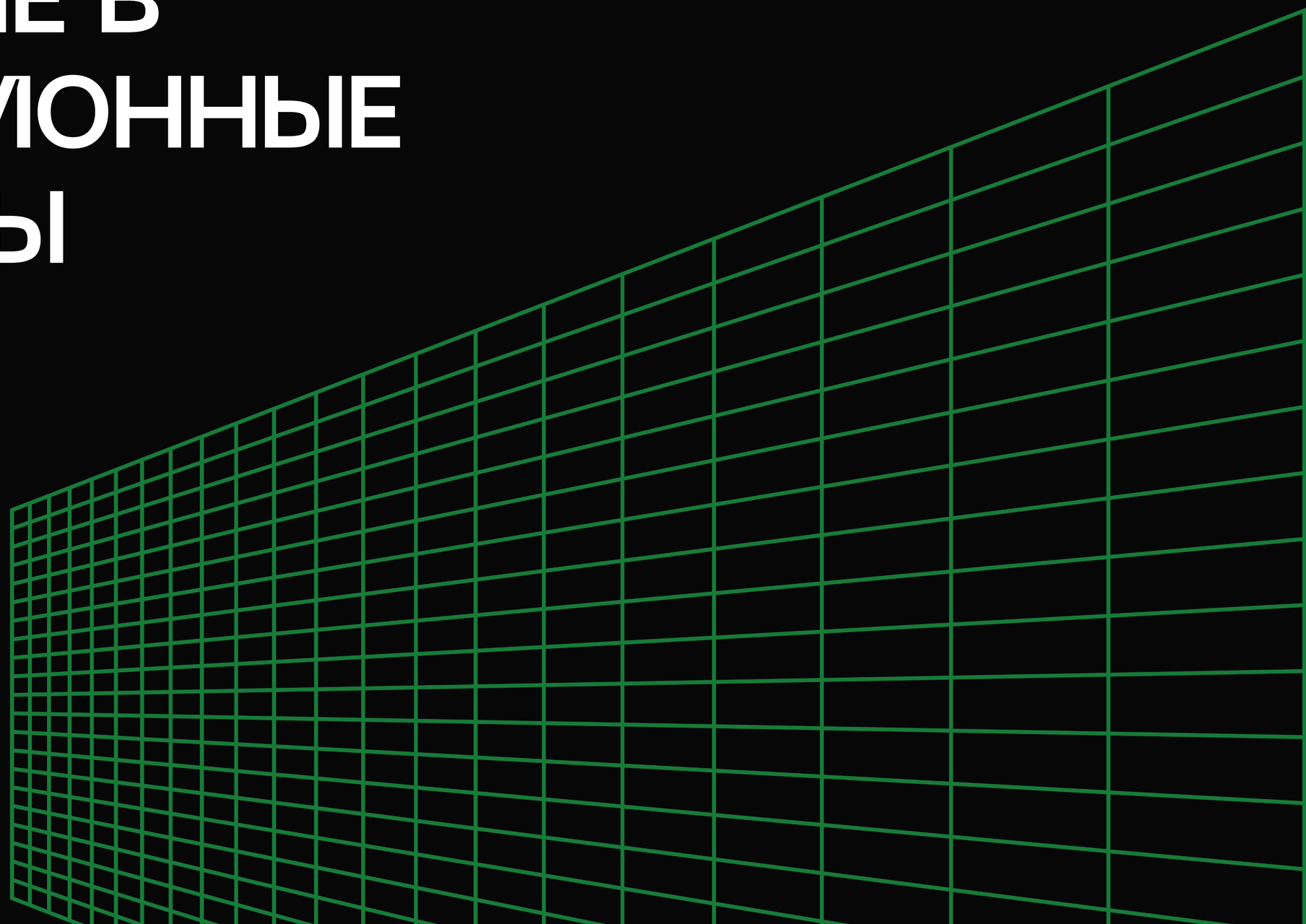
ВВЕДЕНИЕ В ОПЕРАЦИОННЫЕ СИСТЕМЫ



Balun.Courses

Balun.Courses

ЗАПИСЬ

Balun.Courses

ПРАВИЛА ЗАНЯТИЯ

1. Вопросы можно и нужно задавать в любое время, если что-то непонятно
2. Лучше всего задавать вопросы голосом

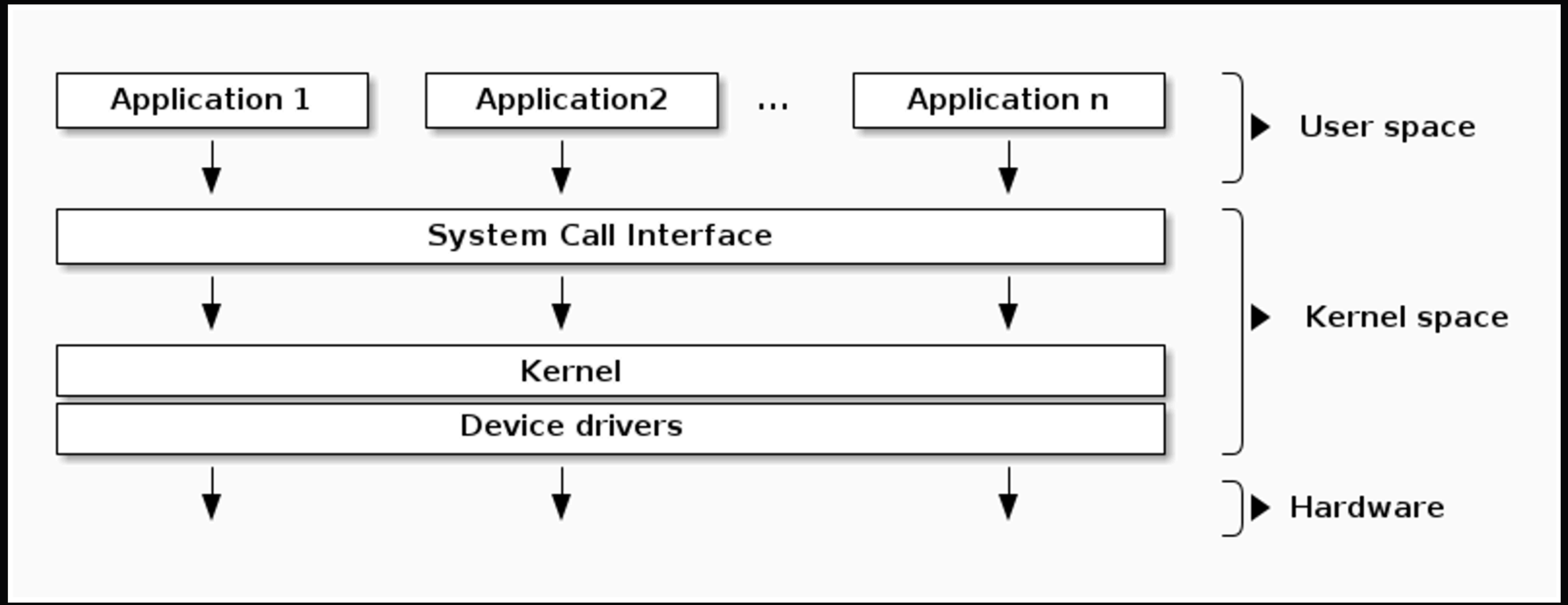
Balun.Courses

ПЛАН ЛЕКЦИИ

1. Системные вызовы
2. Прерывания
3. Сигналы
4. Операционная система xv6
5. Оптимизации использования системных вызовов

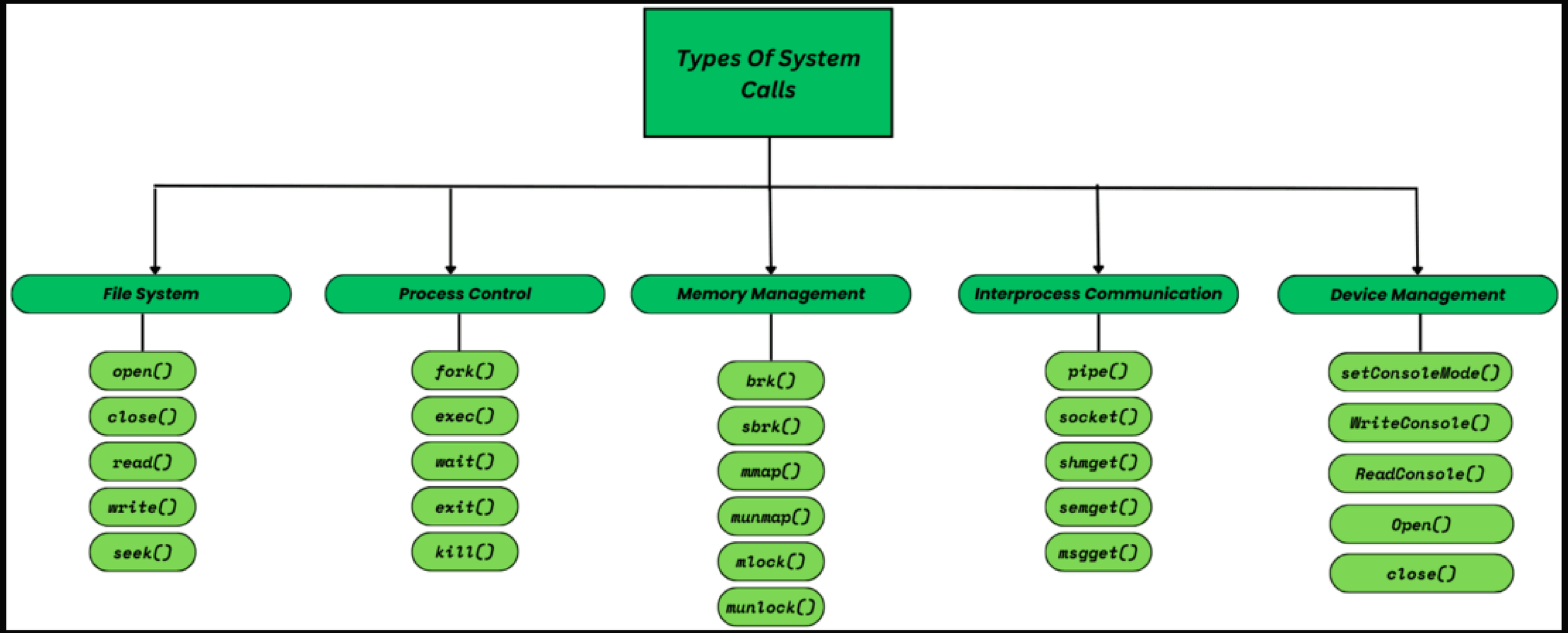
Balun.Courses

ВЗАИМОДЕЙСТВИЕ С OS



Balun.Courses

СИСТЕМНЫЕ ВЫЗОВЫ



Balun.Courses

СИСТЕМНЫЕ ВЫЗОВЫ

%eax	Name	Source	%ebx	%ecx	%edx	%esx	%edi
1	sys_exit	kernel/exit.c	int	-	-	-	-
2	sys_fork	arch/i386/kernel/process.c	struct pt_regs	-	-	-	-
3	sys_read	fs/read_write.c	unsigned int	char *	size_t	-	-
4	sys_write	fs/read_write.c	unsigned int	const char *	size_t	-	-
5	sys_open	fs/open.c	const char *	int	int	-	-
6	sys_close	fs/open.c	unsigned int	-	-	-	-
7	sys_waitpid	kernel/exit.c	pid_t	unsigned int *	int	-	-
8	sys_creat	fs/open.c	const char *	int	-	-	-
9	sys_link	fs/namei.c	const char *	const char *	-	-	-
10	sys_unlink	fs/namei.c	const char *	-	-	-	-
11	sys_execve	arch/i386/kernel/process.c	struct pt_regs	-	-	-	-
12	sys_chdir	fs/open.c	const char *	-	-	-	-
13	sys_time	kernel/time.c	int *	-	-	-	-
14	sys_mknod	fs/namei.c	const char *	int	dev_t	-	-
15	sys_chmod	fs/open.c	const char *	mode_t	-	-	-
16	sys_lchown	fs/open.c	const char *	uid_t	gid_t	-	-
18	sys_stat	fs/stat.c	char *	struct __old_kernel_stat *	-	-	-
19	sys_lseek	fs/read_write.c	unsigned int	off_t	unsigned int	-	-
20	sys_getpid	kernel/sched.c	-	-	-	-	-
21	sys_mount	fs/super.c	char *	char *	char *	-	-
22	sys_oldumount	fs/super.c	char *	-	-	-	-
23	sys_setuid	kernel/sys.c	uid_t	-	-	-	-
24	sys_getuid	kernel/sched.c	-	-	-	-	-
25	sys_stime	kernel/time.c	int *	-	-	-	-
26	sys_ptrace	arch/i386/kernel/ptrace.c	long	long	long	long	-
27	sys_alarm	kernel/sched.c	unsigned int	-	-	-	-
28	sys_fstat	fs/stat.c	unsigned int	struct __old_kernel_stat *	-	-	-
29	sys_pause	arch/i386/kernel/sys_i386.c	-	-	-	-	-
30	sys_utime	fs/open.c	char *	struct utimbuf *	-	-	-

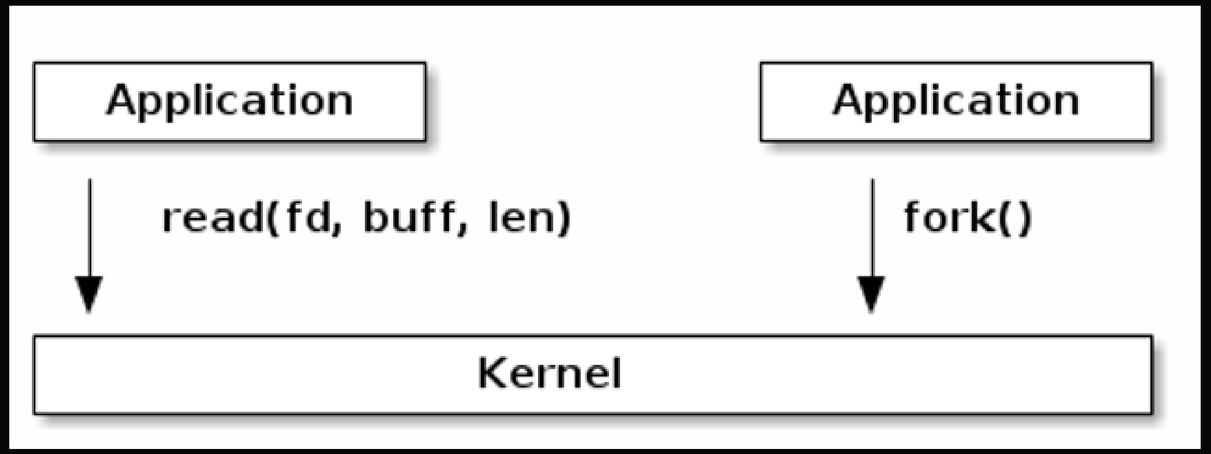
Balun.Courses

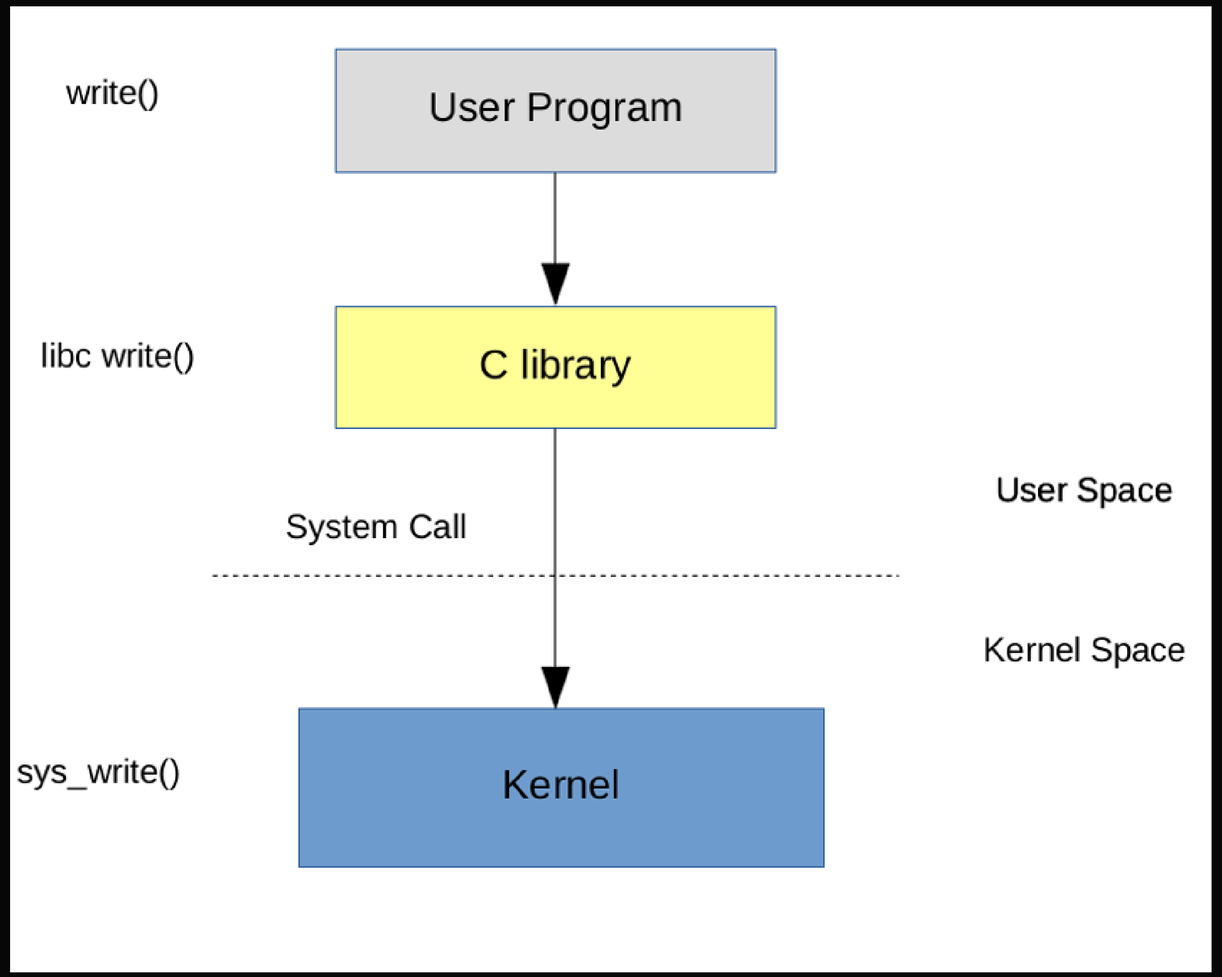
СИСТЕМНЫЕ ВЫЗОВЫ В GO

```go
func main() {
    f, err := os.Open( name: "file.txt")
    fd, err := syscall.Open( path: "file.txt", syscall.O_RDONLY, uint32(os.ModePerm))
    newF, err := unix.Open( path: "file.txt", unix.O_RDONLY, uint32(os.ModePerm))
```

Balun.Courses

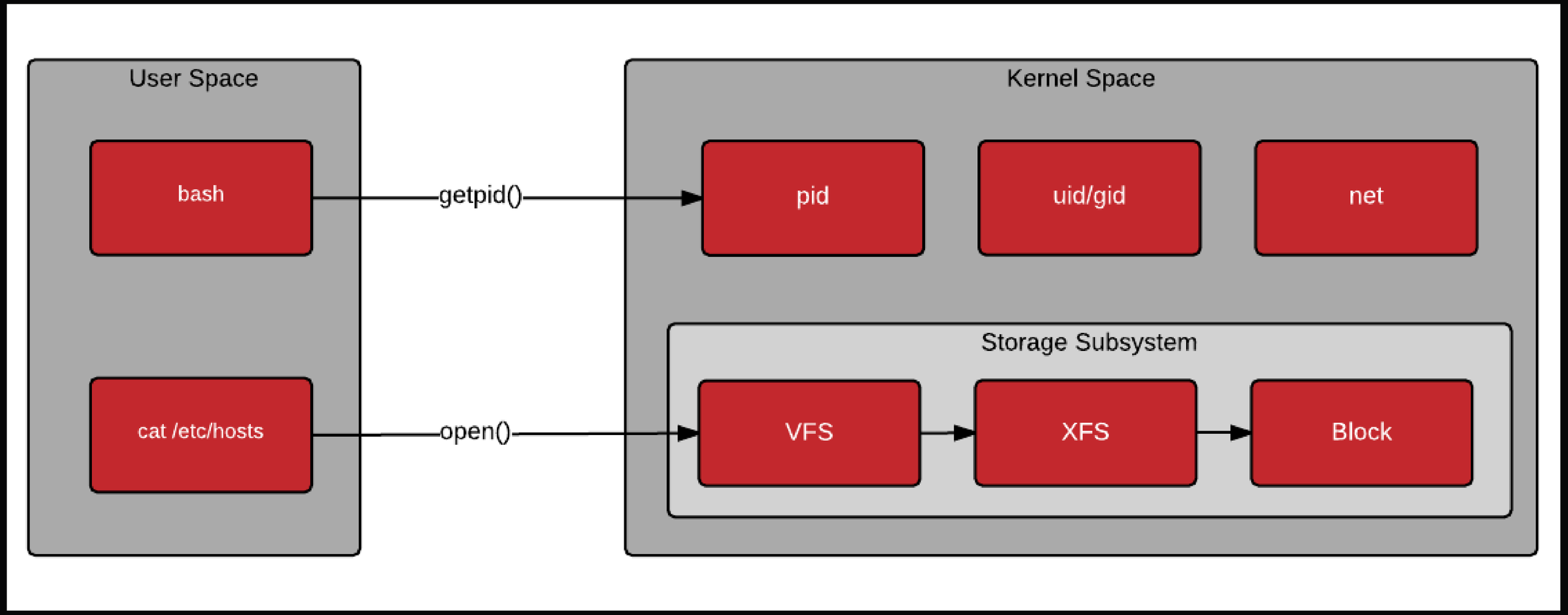
СИСТЕМНЫЕ ВЫЗОВЫ





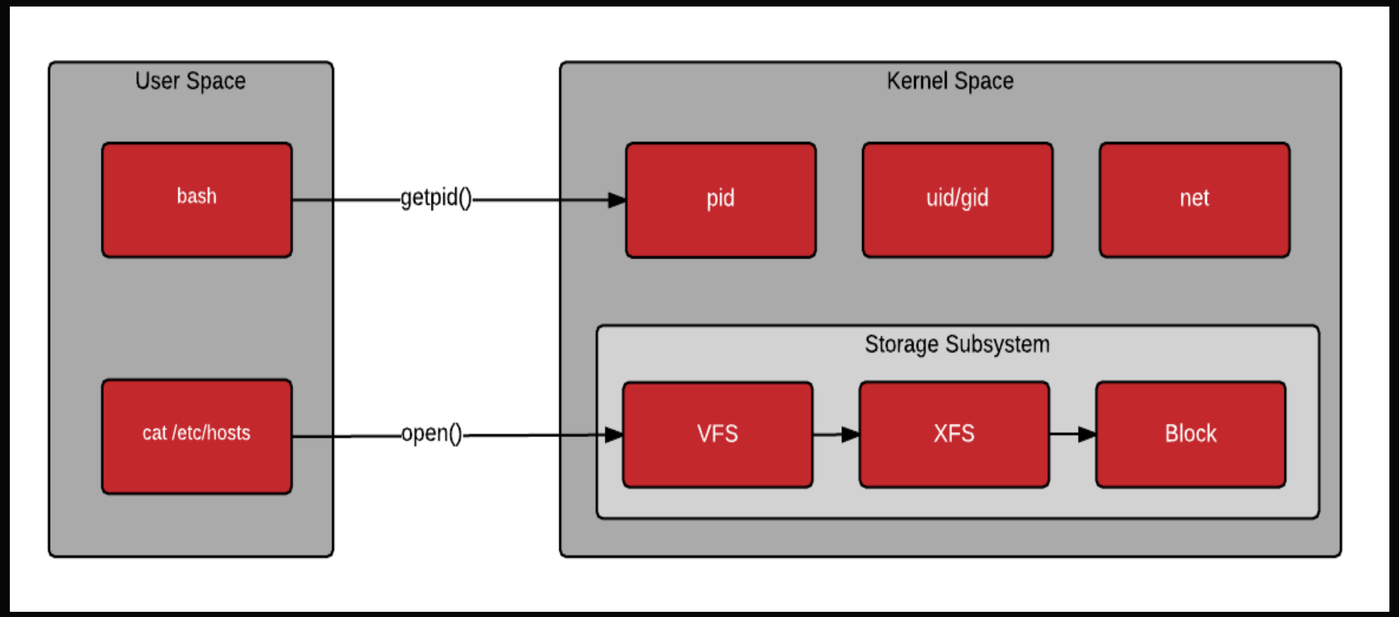
Balun.Courses

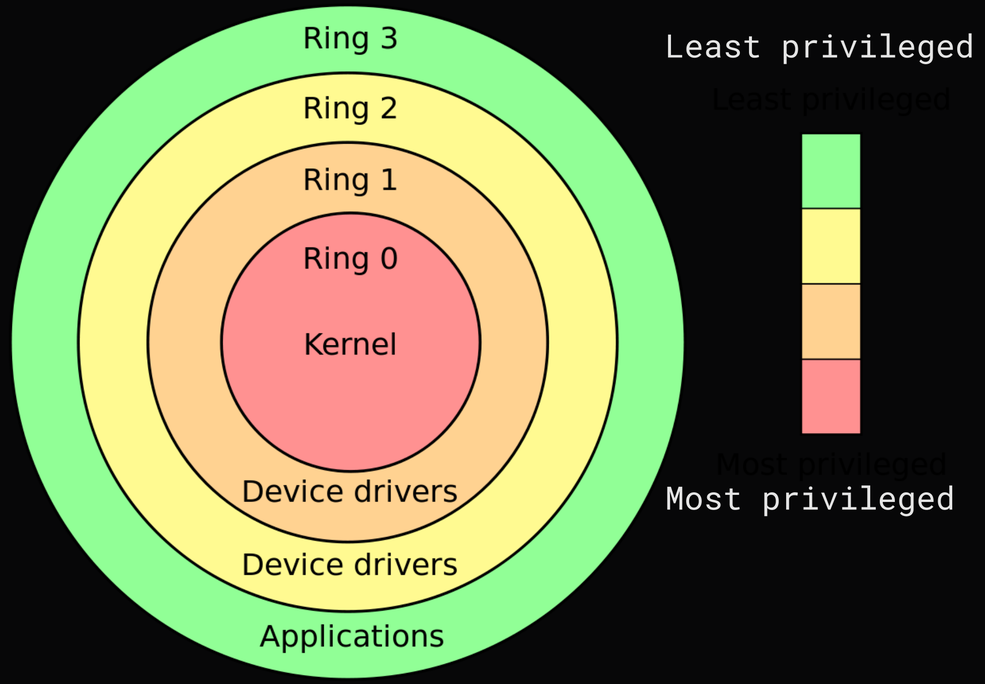
ВЗАИМОДЕЙСТВИЕ С OS



Balun.Courses

ВЗАИМОДЕЙСТВИЕ С OS

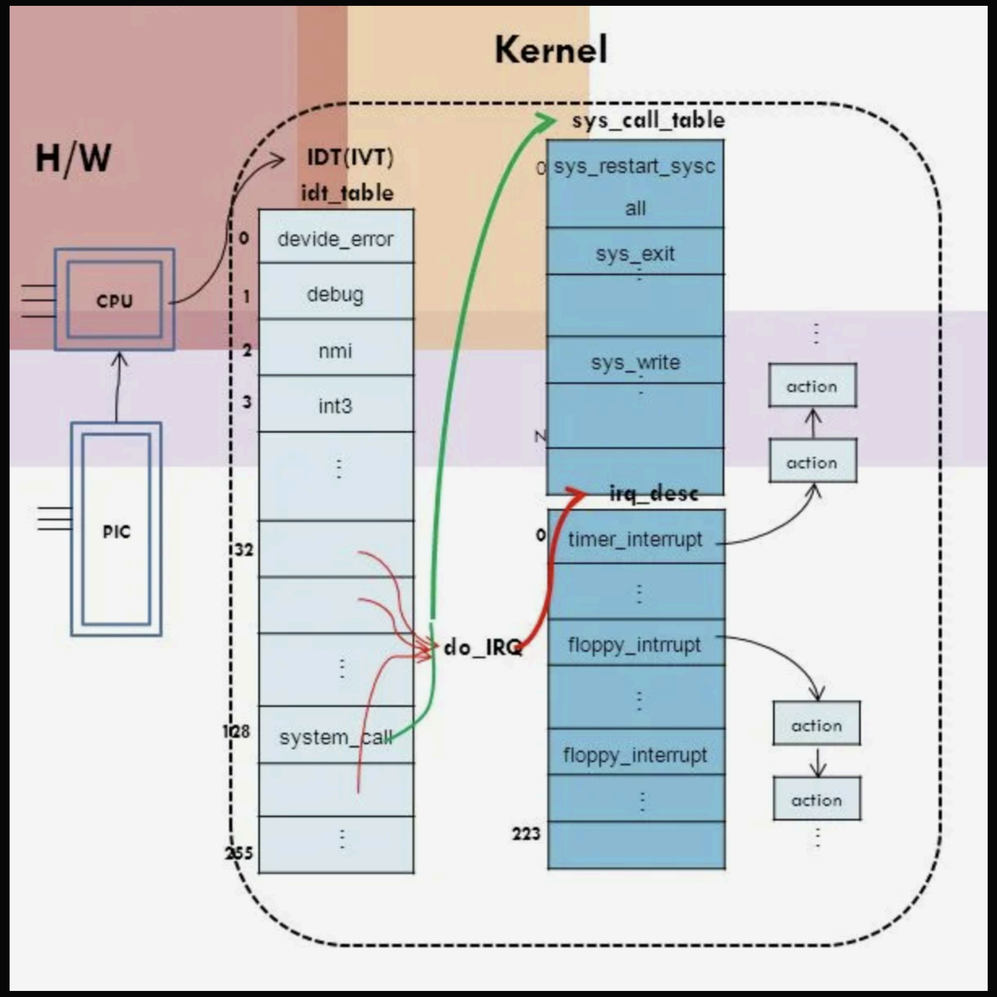




Как именно происходит переключение в kernel space с исполнением кода ядра?

Balun.Courses

ВЗАИМОДЕЙСТВИЕ С OS



IDT - interrupt descriptor table

Как процессор начинает исполнять код ядра?

Balun.Courses

ВЗАИМОДЕЙСТВИЕ С OS

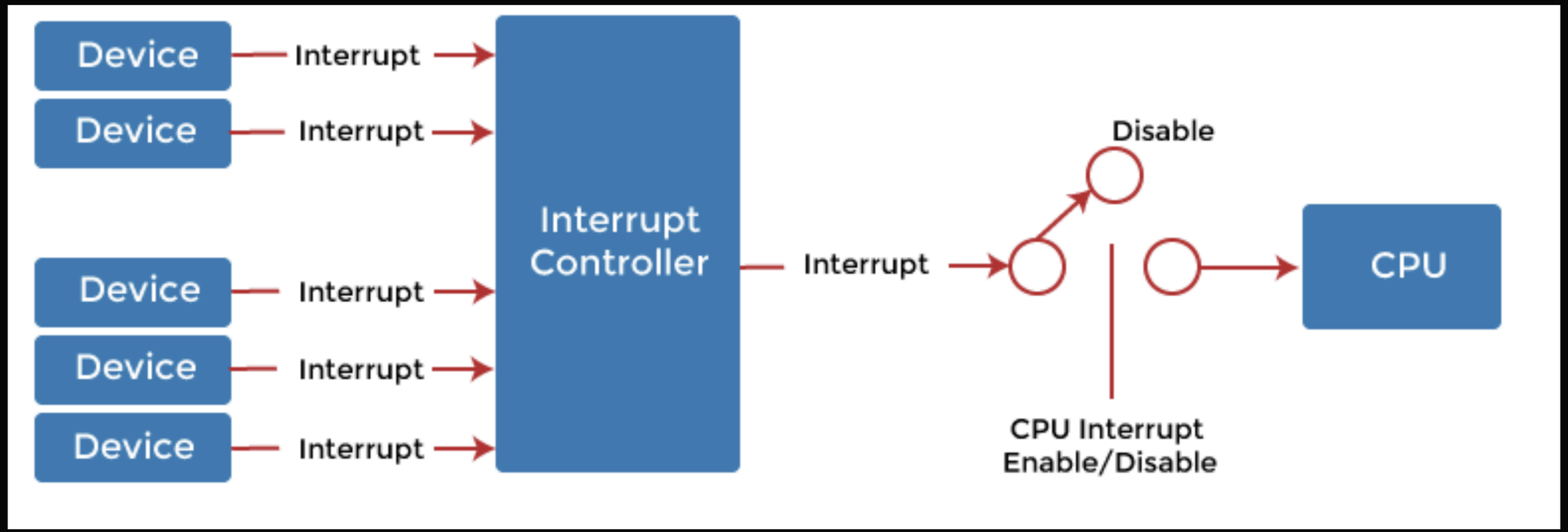
```
func main() {
    a := 1      // 0x6e20dde2
    b := 2      // 0x6e20bbe2
    c := a + b // 0x5a20ddd3
```

PC

PC - program counter

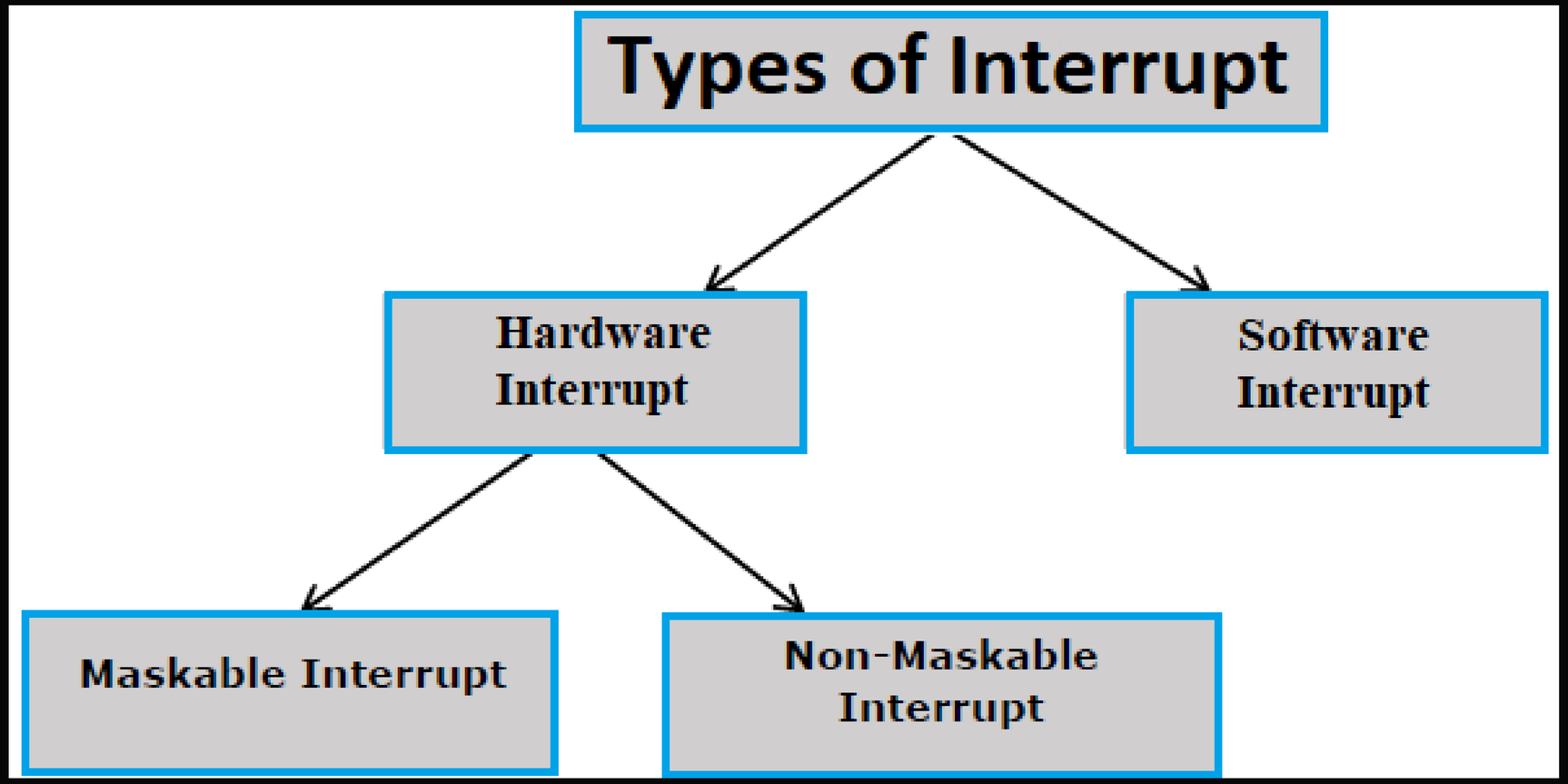
Balun.Courses

ПРЕРЫВАНИЯ



Balun.Courses

ПРЕРЫВАНИЯ

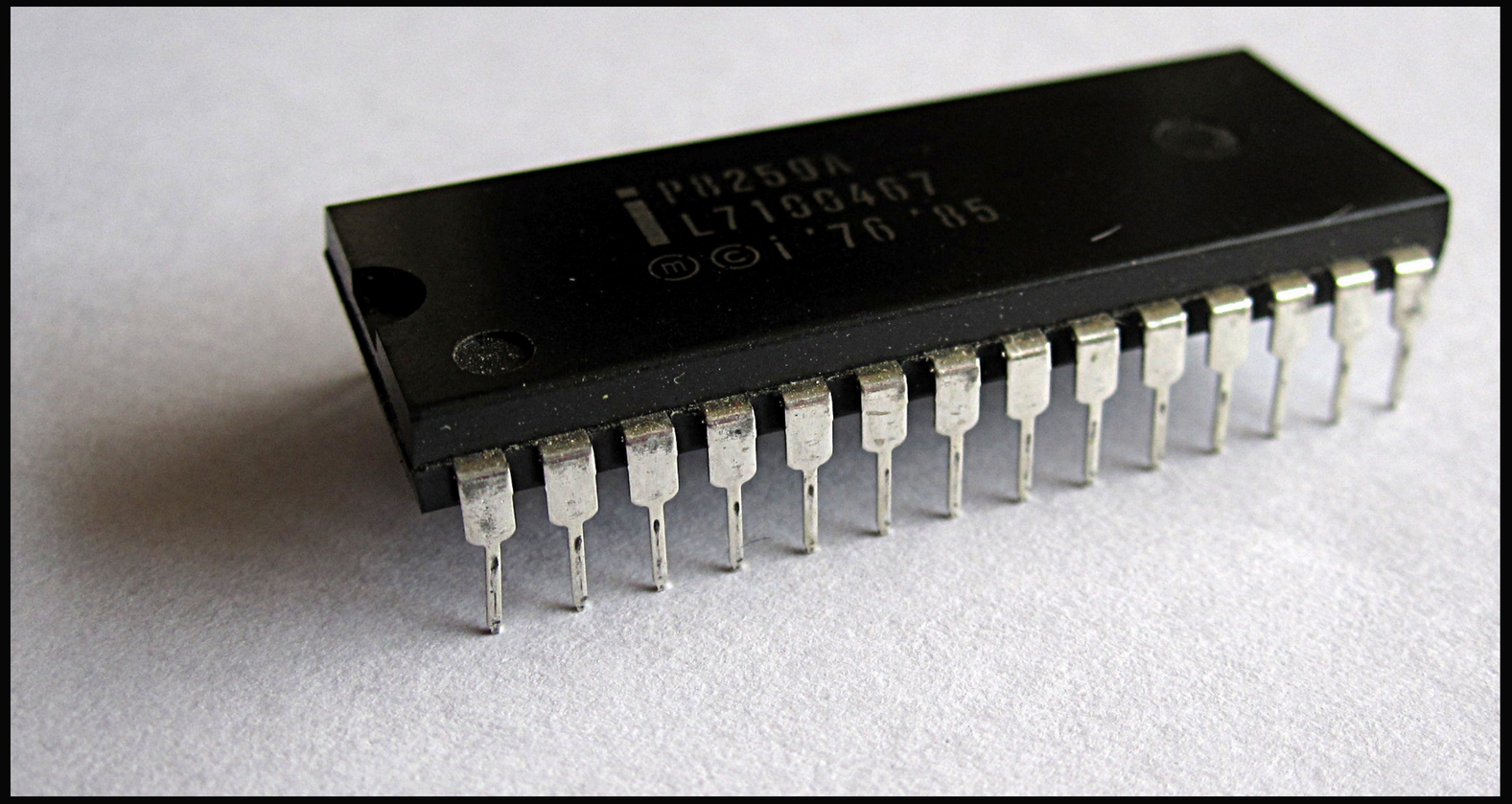


В чем отличия?

Могут использоваться искусственно, например, синхронное прерывание для реализации stop the world GC

Balun.Courses

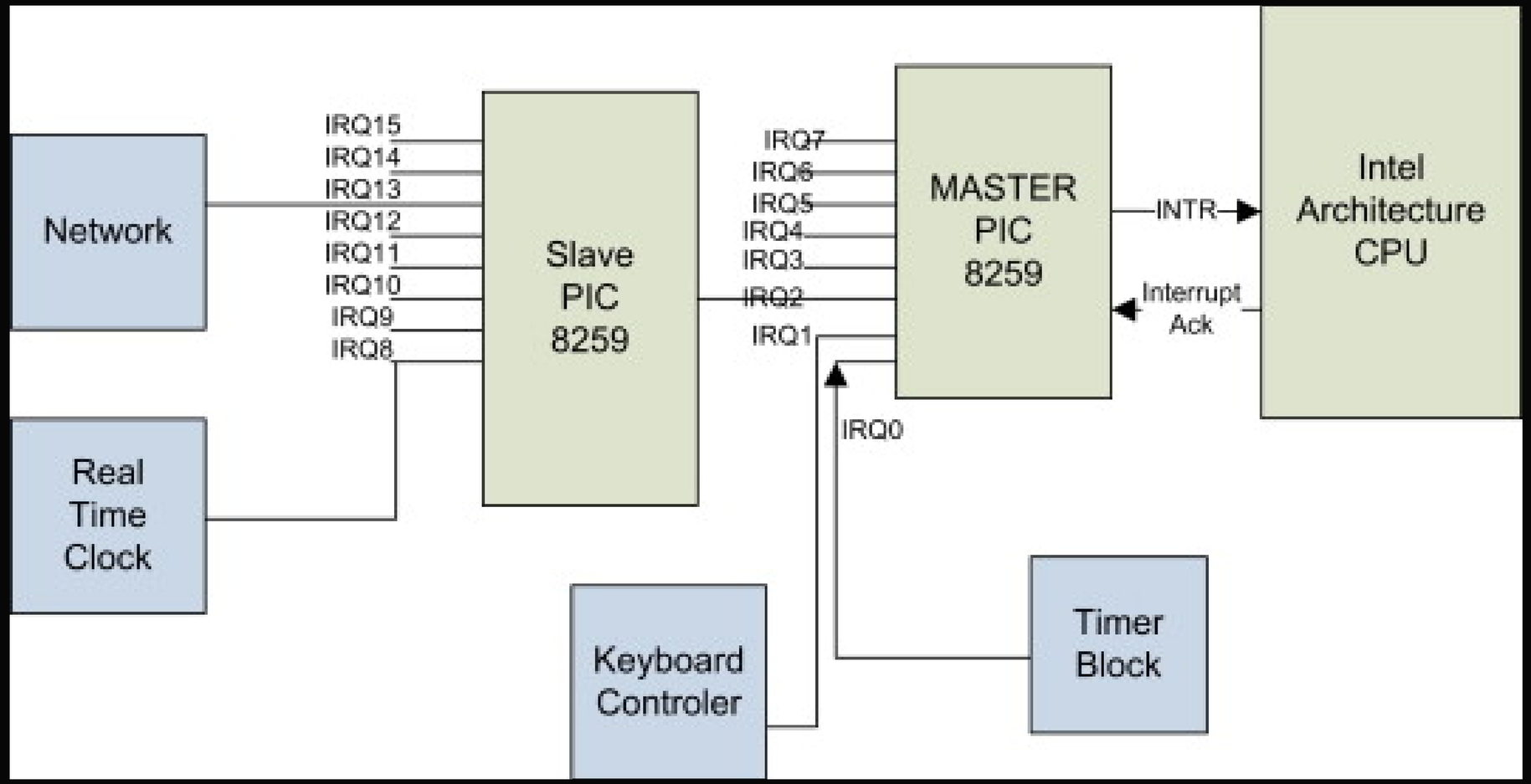
ПРЕРЫВАНИЯ



Intel 8259 PIC

Balun.Courses

ПРЕРЫВАНИЯ



Interrupt request register

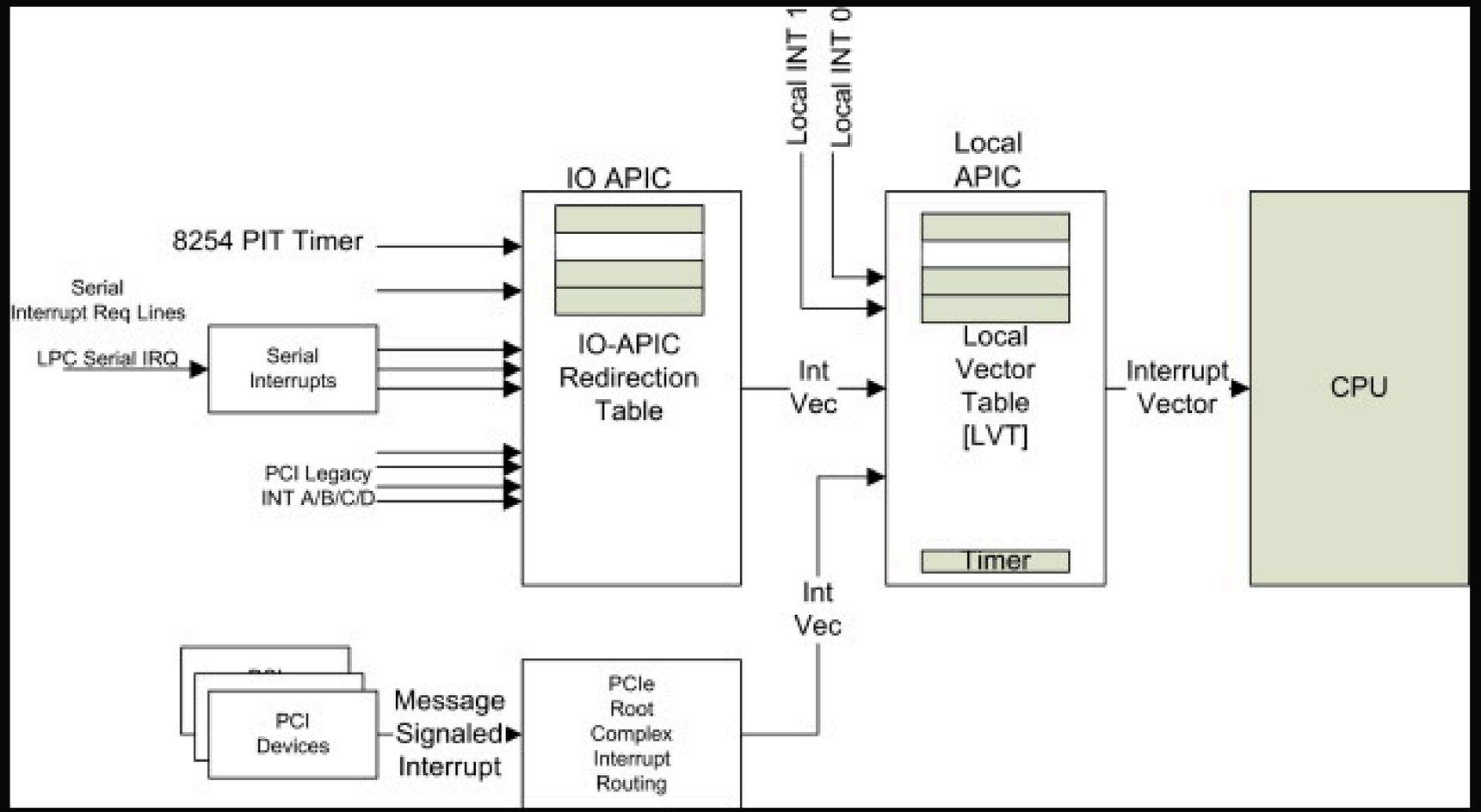
Interrupt mask register

Interrupt in-service register

Legacy Programmable Interrupt Controller

Balun.Courses

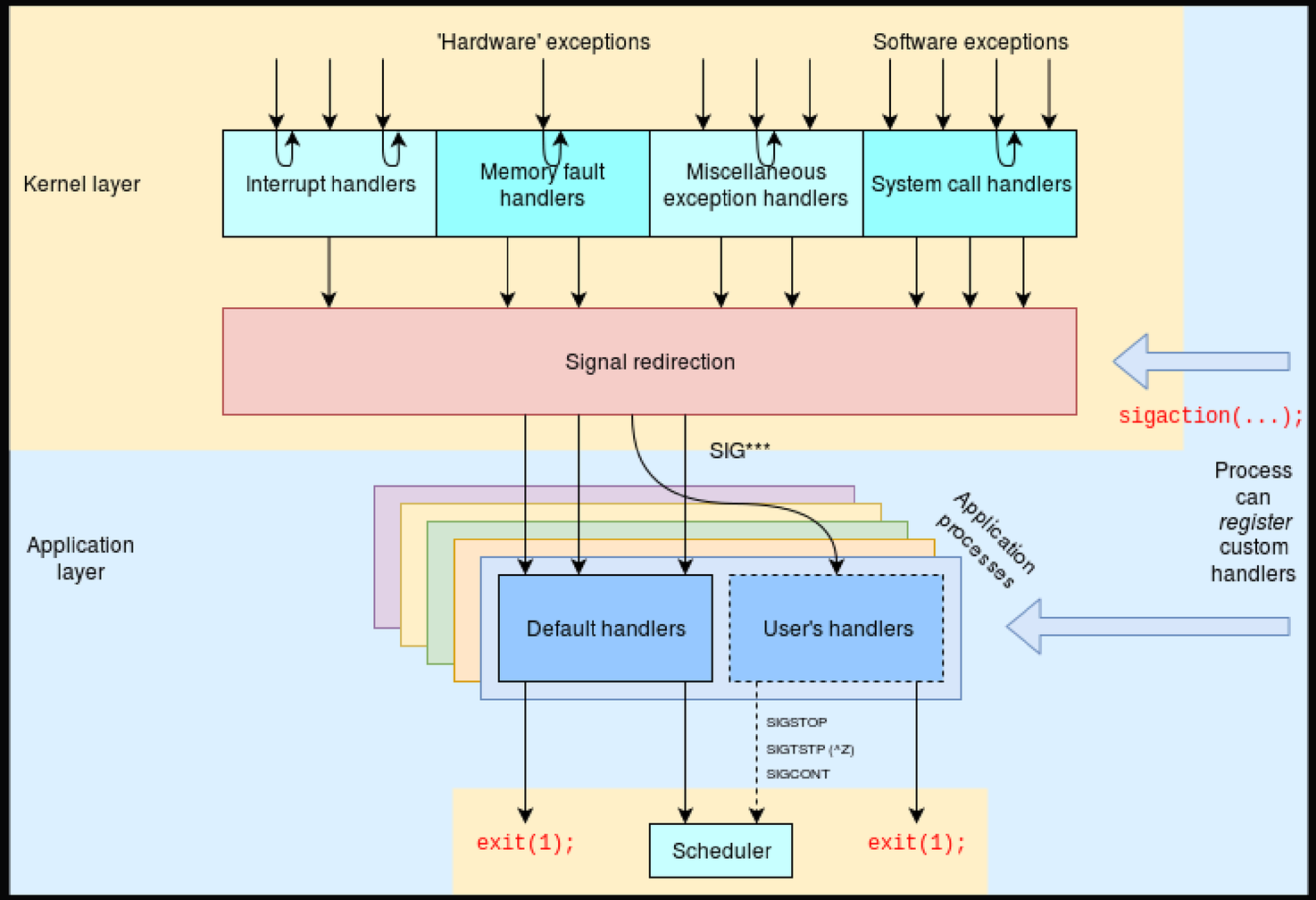
ПРЕРЫВАНИЯ



Advanced Programmable Interrupt Controller

Balun.Courses

СИГНАЛЫ



Linux example

Как зарегистрировать обработчик сигналов в Go?

Balun.Courses

СИГНАЛЫ

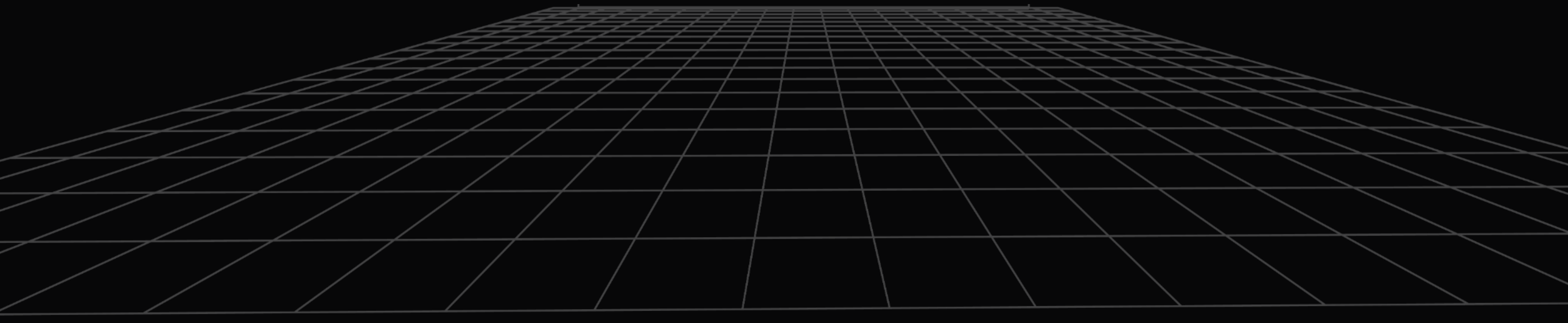
```go
import (
    "context"
    "os/signal"
    "syscall"
    "time"
)

func main() {
    ctx, cancel := signal.NotifyContext(context.Background(), syscall.SIGTERM, syscall.SIGINT, syscall.SIGALRM)
    defer cancel()

    <-ctx.Done()

    // Graceful shutdown, for example
    time.Sleep(time.Second)
}
```

Какие сигналы обычно обрабатывают в user space?

DEMO

Операционная система xv6

Внутреннее устройство



DEMO

Пример устранения избыточных системных вызовов

на примере IO