



```

Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[30103]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62134
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64242
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63397
Sep 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:53:39 amd64 sshd[31269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
Sep 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:23:21 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 01:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 6192.1
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6606]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62078
Sep 24 01:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Sep 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 13:49:08 amd64 sshd[21977]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_midi_event: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_oss: unsupported module, tainting kernel.
Sep 24 20:26:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Sep 24 20:26:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:02 amd64 /usr/sbin/cron[1884]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 25 02:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11686]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778

```

# Practice III: Linux kernel

## Process list (1/10)

Kernel does not differentiate between processes and threads.

- doubly chained, circular list
- each entry of type `struct task_struct`
- type defined in `include/linux/sched.h`
- contains all the information the kernel needs
- `task_struct` definition 132 lines long!
- maximal PID: 32767 (short int)

## Process list (2/10)

```

struct task_struct {
    volatile long state; /* -1 unrunnable, 0 runnable, >0 stopped */

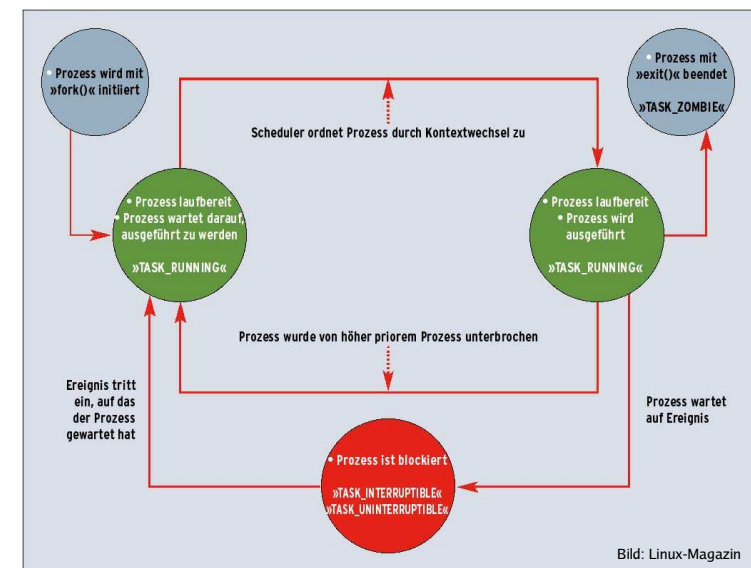
#define TASK_RUNNING      0    executable – is running oder wants to („ready“)
#define TASK_INTERRUPTIBLE 1    sleeps – will wake up when receiving signal
                                or when a certain condition occurs (Kernel)
#define TASK_UNINTERRUPTIBLE 2    sleeps – as above, but no signals
#define TASK_STOPPED      4    doesn't and cannot run (after signals
                                SIGSTOP, SIGTSTP, SIGTIN, SIGTTOU)

#define TASK_TRACED        8
#define TASK_ZOMBIE       16    terminated, but parent process has not yet
                                called wait().

#define TASK_DEAD          32

```

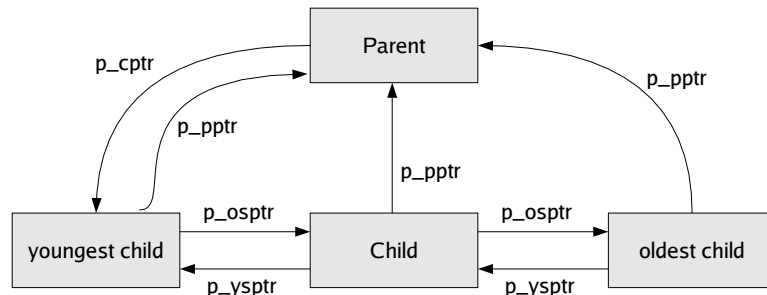
## Process list (3/10)



## Process list (4/10)

### Relationships (old Linux version)

```
struct task_struct {
    [...]
    struct task_struct *p_opptr, *p_pptr, *p_cptr, *p_ysptr, *p_osptr;
}
```



## Process list (6/10)

### Process groups and sessions

```
struct task_struct {
    [...]
    struct task_struct *group_leader;
    /* threadgroup leader */
    [...]
    /* signal handlers */
    struct signal_struct *signal;
}

struct signal_struct {
    /* job control IDs */
    pid_t pgrp;      Process Group ID
    pid_t tty_old_pgrp;
    pid_t session;  Session ID
    /* boolean value for session
    group leader */
    int leader;
}
```

- Each process is member of a process group
- Process Group ID (PGID) – `ps j`
- `current->signal->pgrp`

## Process list (5/10)

### Relationships (new Linux version)

```
struct task_struct {
    [...]
    struct task_struct *parent; /* parent process */
    struct list_head children; /* list of my children */
    struct list_head sibling; /* linkage in my parent's children list */
}
```

#### Code for accessing all children:

```
list_for_each(list, &current->children) {
    task = list_entry(list, struct task_struct, sibling);
    /* task now points to one of the children */
}
```

#### Walk process tree up to the root, i.e. init:

```
for (task = current; task != &init_task; task = task->parent) {
    ...
}
```

## Process list (7/10)

### Process groups

- Signals to all members of a process group:  
`killpg(pgrp, sig);`
- Waiting for a child of your own process group:  
`waitpid(0, &status, ...);`
- ... or some other, specific process group:  
`waitpid(-pgrp, &status, ...);`

## Process list (8/10)

### Sessions

- Typically a new session starts with a login shell
- All processes started from this shell belong to the (same) session
- Common „controlling TTY“

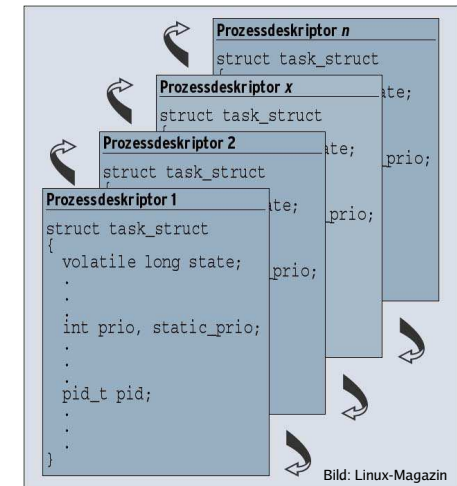
## Process list (10/10)

### Walk through process list

```
struct list_head tasks;
```

```
#define next_task(p)
    list_entry((p)->tasks.next,
               struct task_struct, tasks)
#define prev_task(p)
    list_entry((p)->tasks.prev,
               struct task_struct, tasks)
```

```
#define for_each_process(p)
    for (p = &init_task ;
         (p = next_task(p))
         != &init_task ; )
```



## Process list (9/10)

```
> ps j
PPID  PID  PGID  SID  TTY      TPGID  STAT  UID   TIME COMMAND
19287  7628  7628  19287 pts/8    19287  S      500   0:00 /bin/sh /usr/bin/mozilla -mail
7628  7637  7628  19287 pts/8    19287  Sl     500   20:50 /opt/moz/lib/mozilla-bin -mail
9634  10095 10095 10095 tty1     10114  Ss     500   0:00 -bash
10095  10114 10114 10095 tty1     10114  S+     500   0:00 /bin/sh /usr/X11R6/bin/startx
10095  10115 10114 10095 tty1     10114  S+     500   0:00 tee /home/esser/.X.err
10114  10135 10114 10095 tty1     10114  S+     500   0:00 xinit /home/esser/.xinitrc
10135  10151 10151 10095 tty1     10114  S      500   0:00 /bin/sh /usr/X11R6/bin/kde
10151  10238 10151 10095 tty1     10114  S      500   0:00 kwrapper kmsserver
10258  10270 10270 10270 pts/2    10270  Ss+    500   0:00 bash
10276  10278 10278 10278 pts/4    10278  Ss+    500   0:00 bash
10260  10284 10284 10284 pts/5    10284  Ss+    500   0:00 bash
10275  10292 10292 10292 pts/6    10989  Ss     500   0:00 bash
10259  10263 10263 10263 pts/1    10263  Ss+    500   0:00 bash
10263  28869 28869 10263 pts/1    10263  S      500   0:16 konqueror /media/usbdisk/dcim
10263  28872 28872 10263 pts/1    10263  S      500   0:13 konqueror /home/esser
29201  29203 29203 29203 pts/7    29203  Ss+    500   0:00 bash
4822  4823  4823  4823 pts/14   4823  Ss+    500   0:00 -bash
4823  31118 31118 4823  pts/14   4823  S      500   0:00 nedit kernel/sched.c
4823  31297 31297 4823  pts/14   4823  S      500   0:00 nedit kernel/fork.c
23115  32703 32703 23115 pts/13   32703  R+     500   0:00 ps j
```

## Process creation (1/2)

Most important file in the kernel sources: kernel/fork.c  
(contains copy\_process)

- fork() calls clone(),
- clone() calls do\_fork(), and
- do\_fork() calls copy\_process(), wherein:

## Process creation (2/2)

copy\_process() does:

- dup\_task\_struct(): new kernel stack, thread\_info structure, task\_struct-entry
- sets child status to TASK\_UNINTERRUPTIBLE
- copy\_flags(): PF\_FORKNOEXEC
- get\_pid(): obtain a fresh PID for child
- Depending on clone() parameters: copy or share open files, signal handlers, process memory areas etc.
- split remaining process time (→ scheduler)

Then: wake up and start (child runs before father)

```
Step 19 14:27:18 amd64 sshd[26494]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61507
Step 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Step 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Step 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 20 02:00:01 amd64 /usr/sbin/cron[31031]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62004
Step 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Step 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62105
Step 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62514
Step 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64262
Step 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Step 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63375
Step 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Step 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63546
Step 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Step 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63397
Step 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Step 21 17:43:39 amd64 sshd[31269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
Step 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Step 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Step 22 01:00:01 amd64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Step 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 22 02:00:01 amd64 /usr/sbin/cron[4891]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 22 20:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Step 23 01:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 23 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from ::ffff:168.1.5 port 59771 ssh2
Step 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Step 23 18:04:34 amd64 sshd[6606]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62093
Step 24 01:00:01 amd64 /usr/sbin/cron[1436]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Step 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Step 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Step 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Step 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Step 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Step 24 15:42:07 amd64 kernel: amd_seq_midi_event: unsupported module, tainting kernel.
Step 24 15:42:07 amd64 kernel: amd_seq_oss: unsupported module, tainting kernel.
Step 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Step 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Step 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Step 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Step 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 25 02:00:02 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c "age > *30d*")
Step 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Step 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Step 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Step 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Step 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Step 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Step 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Step 25 14:06:10 amd64 sshd[11586]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Step 25 14:07:17 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Step 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Step 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778
```

## 3. Interrupts (1/2)

## Threads in the kernel

- Linux has no concept of threads, but views them as processes
- Thread: process that shares several resources with other processes
- Every thread has task\_struct and looks like a normal process (to the kernel)
- Fundamentally different from e.g. Windows and Solaris

## Interrupts: Introduction

### Interrupt classes

- **Software Interrupts (Exception, synchronous Interrupt)**  
False memory access, Division by 0, false CPU instruction, ...
- **Timer**
- **I/O (Input/Output, asynchr. Interrupt)**  
From the I/O-Controller: Action completed
- **Hardware error**  
Power failure, RAM parity error

# Interrupts: Introduction

## Why Interrupts?

- **Efficiency**

I/O access very slow → long waiting times, when processes wait until termination of I/O

- **Programming logic**

Don't query device status over and over again, but wait for Interrupt

- **Kein Polling**

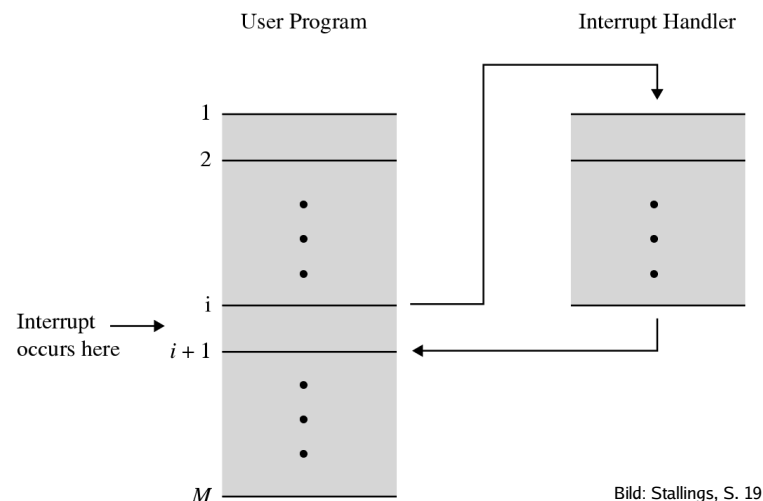
Polling: OS regularly queries all devices, whether there was an event

# Interrupts: Introduction

## Basic procedure

- Interrupt occurs
- Running thread is interrupted (after last instruction), OS takes control
- OS saves thread data (as in a task switch → scheduler)
- OS calls Interrupt handler
- Then: Scheduler chooses process which can continue (e.g. the interrupted one)

# Interrupts: Introduction



# Interrupts: Introduction

## What to do in case of multiple interrupts?

Three possibilities:

- During processing of an interrupt: disable all others (DI, disable interrupts)  
→ Interrupt queue
- Allow other interrupts while processing one
- Interrupt priorities: Only interrupts with higher priority can interrupt those with a lower one

# Interrupts: Introduction

## What to do in case of multiple interrupts?

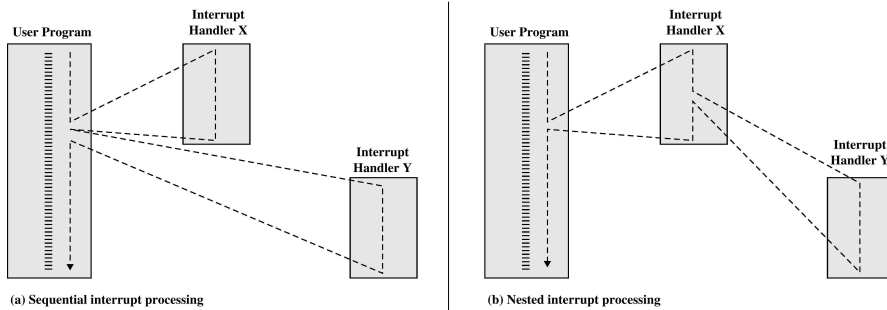


Bild: Stallings, S. 26

# Practice: Interrupts in Linux

```

Sep 19 14:27:18 amd64 sshd[26494]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61507
Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[10103]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[616]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62514
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64242
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63397
Sep 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:53:39 amd64 sshd[31269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
Sep 21 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[24888]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[14649]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:23:21 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 01:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[25555]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from ::ffff:192.168.1.100 port 62530
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:34 amd64 sshd[6066]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64596
Sep 24 01:00:01 amd64 /usr/sbin/cron[14336]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Sep 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: amd_seq_ops: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 kernel: amd_seq_ops: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Sep 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c "age > *30d")
Sep 25 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11586]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778
    
```

# Interrupts: Introduction

## Multitasking and Interrupts

- Multitasking improves CPU utilization:
  - I/O-heavy process waits for I/O events,
  - CPU-heavy process goes on calculating
- Process starts I/O operation and goes to sleep (waits for signal)

# Practice

```

> cat /proc/interrupts
CPU0
0: 3353946487          XT-PIC  timer
2:                    0          XT-PIC  cascade
3:                    4663         XT-PIC  NVidia CK804
5: 159275991          XT-PIC  ohci1394, nvidia
7:                    971775         XT-PIC  hsfpcibasic2
8:                    2          XT-PIC  rtc
9:                    0          XT-PIC  acpi
10:                   31052         XT-PIC  libata, ohci_hcd
11: 197906977         XT-PIC  libata, ehci_hcd
12: 16904921          XT-PIC  eth0
14: 60349322          XT-PIC  ide0
NMI: 0
LOC: 0
ERR: 0
MIS: 0
    
```

# Interrupt Handler

## For each device:

- Interrupt Request (IRQ) Line
- Interrupt Handler (Interrupt Service Routine, ISR) → part of the device driver
- C function
- runs in special context (Interrupt Context)
- „top half“ and „bottom half“

# Treiberprogrammierung

## Treiber registrieren mit Interrupt handler:

```
int request_irq(  
    unsigned int irq,                /* Welche IRQ-Nummer? */  
    irqreturn_t (*handler)(int, void *, struct pt_regs *),  
    unsigned long irqflags,  
    const char * devname,           /* Geräte-name->/proc/int..*/  
    void *dev_id);
```

- Interrupt mit IRQ `irq` wird ausgelöst
- BS ruft Interrupt handler `handler()` auf
- Flags:
  - SA\_SHIRQ: Interrupt für mehrere Treiber
  - SA\_INTERRUPT: Lokale Interrupts werden gesperrt
  - SA\_SAMPLE\_RANDOM: Interrupts treten „zufällig“ auf, nutzen; Entropie-Vergrößerung, Zufallszahlen

# top and bottom half

## top half

- Interrupt handler
- starts immediately, handles time-critical things
- acknowledges (to the device) reception of interrupt, resets device, etc.
- Everything else → bottom half

## bottom half

- starts delayed, does most of the work
- more about it: next lecture