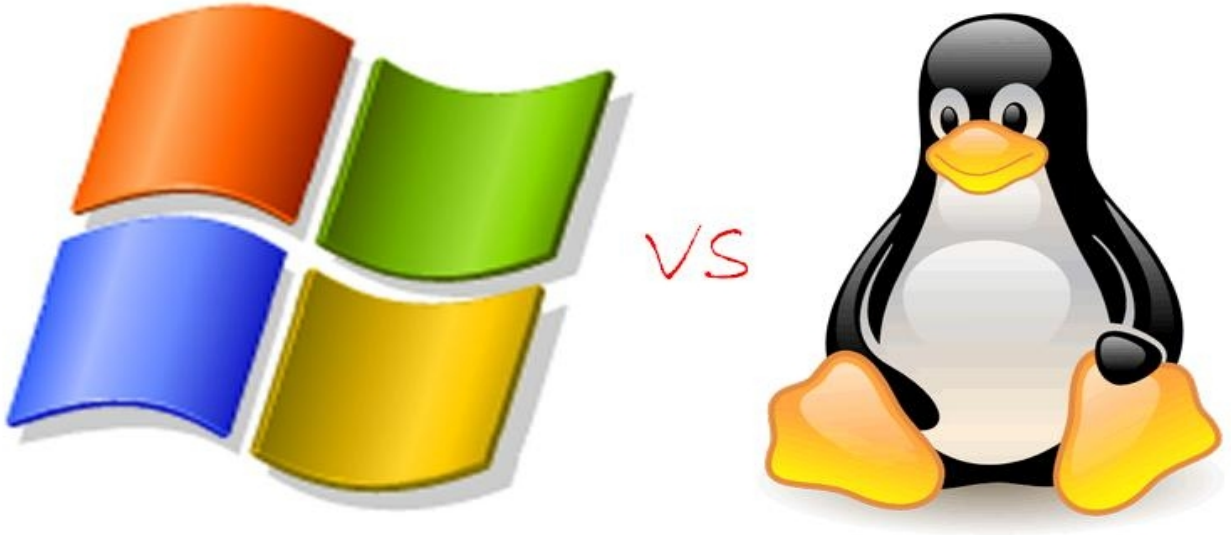
Windows vs Linux:
kernel a confronto



Linux Day 2010

Fermo – 23 ottobre 2010

Andrea Colangelo

Ubuntu Developer


CAUTION CAUTION CAUTION

A problem has been detected and Windows has been shut down to prevent damage
to your computer.

TRAP_CAUSE_UNKNOWN

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

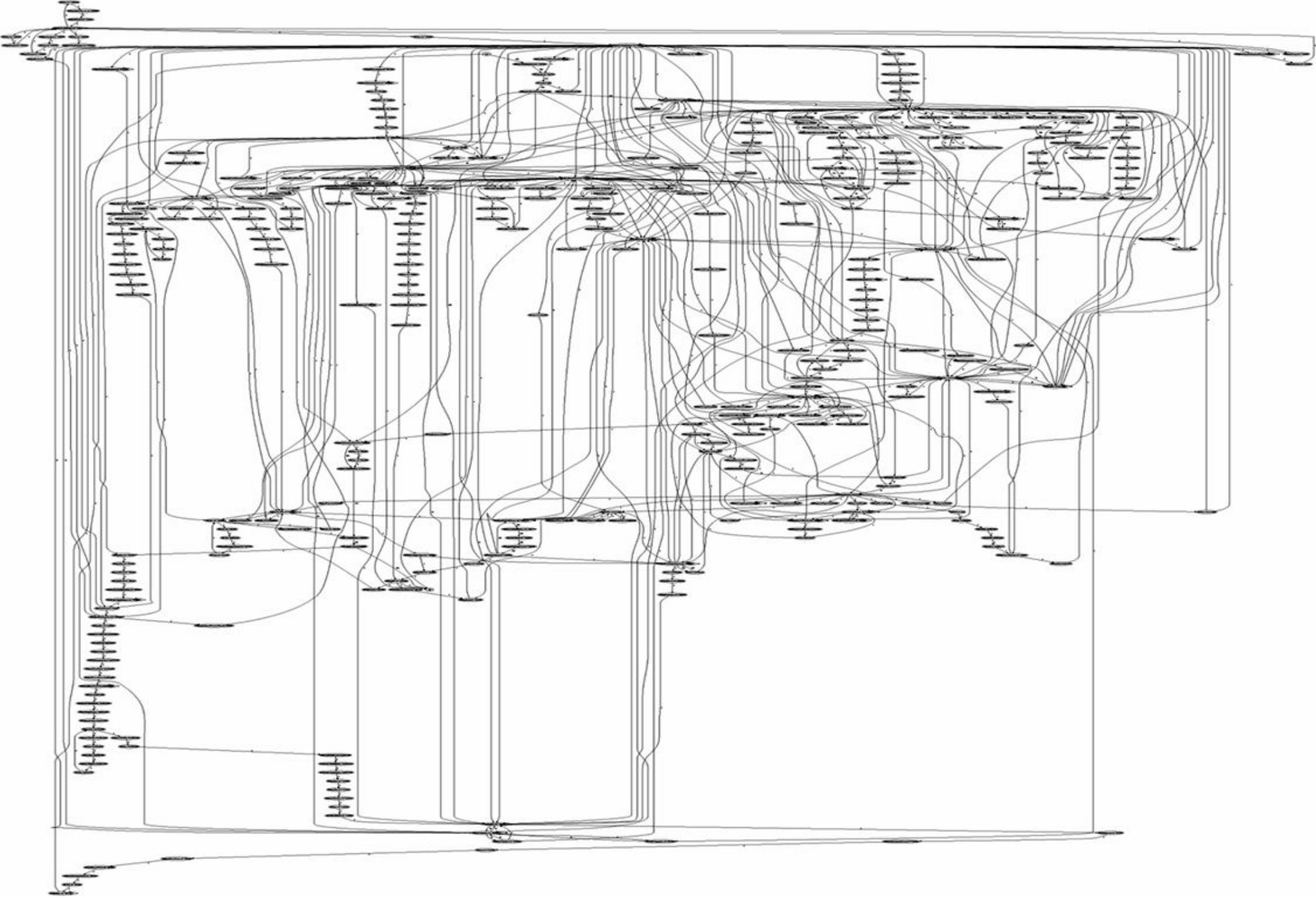
Technical information:

*** STOP: 0x00000012 (0x00000000,0x00000000,0x00000000,0x00000000)

Collecting data for crash dump ...
Initializing disk for crash dump ...
Beginning dump of physical memory.
Dumping physical memory to disk: 60

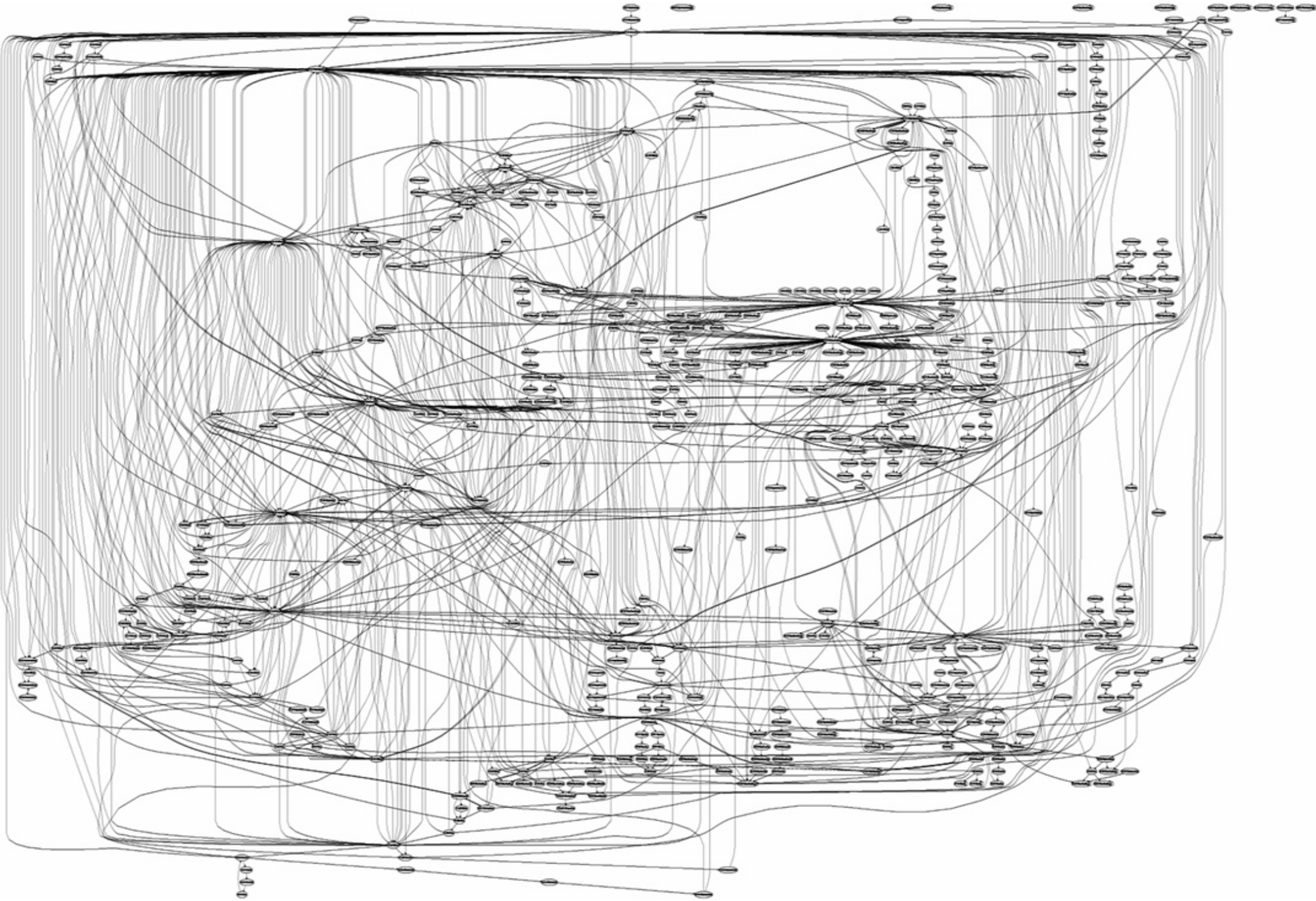
```
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
BUG: unable to handle kernel paging request at virtual address 78bc78bb
 printing eip:
eabb8e00
*pde = 2fd3a001
*pte = 00000000
Oops: 0002 [#1]
SMP
Modules linked in: drbd ipmi_si ipmi_devintf ipmi_msghandler
CPU:    2
EIP:    0060:[<eabb8e00>]    Not tainted VLI
EFLAGS: 00010286   (2.6.18-pe-sm-tmm-20061030-vs #1)
EIP is at 0xeabb8e00
eax: eabb8e00   ebx: c5833900   ecx: c5927a40   edx: d742a000
esi: eabb8e00   edi: 00000007   ebp: 00000000   esp: d742beec
ds: 007b   es: 007b   ss: 0068
Process readpgenv (pid: 5839[#0], ti=d742a000 task=f08ba030 task.ti=d742a000)
Stack: c0129cc3 eabb8e00 c5833980 00000000 00000008 c0129eaa c011facf 00000000
       00000001 c068c328 00000008 c011f743 c068c328 00000002 00000000 00000046
       b7f8c4f8 c0110910 0805ae70 c011f7fd d742bfbc c01032d7 d742bfbc 0000007b
Call Trace:
 [<c0129cc3>] __rcu_process_callbacks+0xfd/0x162
 [<c0129eaa>] rcu_process_callbacks+0x20/0x40
 [<c011facf>] tasklet_action+0x55/0xb0
 [<c011f743>] __do_softirq+0x79/0xfd
 [<c0110910>] do_page_fault+0x0/0x582
 [<c011f7fd>] do_softirq+0x36/0x3a
 [<c01032d7>] apic_timer_interrupt+0x1f/0x24
 [<c0110910>] do_page_fault+0x0/0x582
 [<c011007b>] early_vga_write+0x51/0x102
 [<c0110a68>] do_page_fault+0x158/0x582
 [<c012421c>] sys_rt_sigaction+0x68/0x7e
 [<c0110910>] do_page_fault+0x0/0x582
 [<c0103389>] error_code+0x39/0x40
Code: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 <c4> 79 65 c9 c9 8f 15 c0 00 00 00 00 00 00 00 00 40 04 60 c0 00
EIP: [<eabb8e00>] 0xeabb8e00 SS:ESP 0068:d742beec
 <0>Kernel panic - not syncing: Fatal exception in interrupt
```

```
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI ioctl, please convert it to SG_IO
program smartd is using a deprecated SCSI io      please convert it to SG_IO
program smartd is using a deprecated SCSI i          ase convert it to SG_IO
program smartd is using a deprecated SCSI              onvert it to SG_IO
program smartd is using a deprecated SCSI                  rt it to SG_IO
BUG: unable to handle kernel paging reque                   c78bb
 printing eip:
eabb8e00
*pde = 2fd3a001
*pte = 00000000
Oops: 0002 [#1]
SMP
Modules linked in: drbd ipmi_si ip
CPU:    2
EIP:    0060:[<eabb8e00>]    Not
EFLAGS: 00010286   (2.6.18-pe-sm
EIP is at 0xeabb8e00
eax: eabb8e00   ebx: c5833900
esi: eabb8e00   edi: 00000007
ds: 007b   es: 007b   ss: 00
Process readpgenv (pid: 5835
Stack: c0129cc3 eabb8e00 c5
       00000001 c068c328 00000
       b7f8c4f8 c0110910 0805ae70
Call Trace:
 [<c0129cc3>] __rcu_process_callbacks+0xfd/0
 [<c0129eaa>] rcu_process_callbacks+0x20/0x40
 [<c011facf>] tasklet_action+0x55/0xb0
 [<c011f743>] __do_softirq+0x79/0xfd
 [<c0110910>] do_page_fault+0x0/0x582
 [<c011f7fd>] do_softirq+0x36/0x3a
 [<c01032d7>] apic_timer_interrupt+0x1f/0x24
 [<c0110910>] do_page_fault+0x0/0x582
 [<c011007b>] early_vga_write+0x51/0x102
 [<c0110a68>] do_page_fault+0x158/0x582
 [<c012421c>] sys_rt_sigaction+0x68/0x7e
 [<c0110910>] do_page_fault+0x0/0x582
 [<c0103389>] error_code+0x39/0x40
Code: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 <c4> 79 65 c9 c9 8f 15 c0 00 00 00 00 00 00 00 00 40 04 60 c0 00
EIP: [<eabb8e00>] 0xeabb8e00 SS:ESP 0068:d742beec
 <0>Kernel panic - not syncing: Fatal exception in interrupt
```

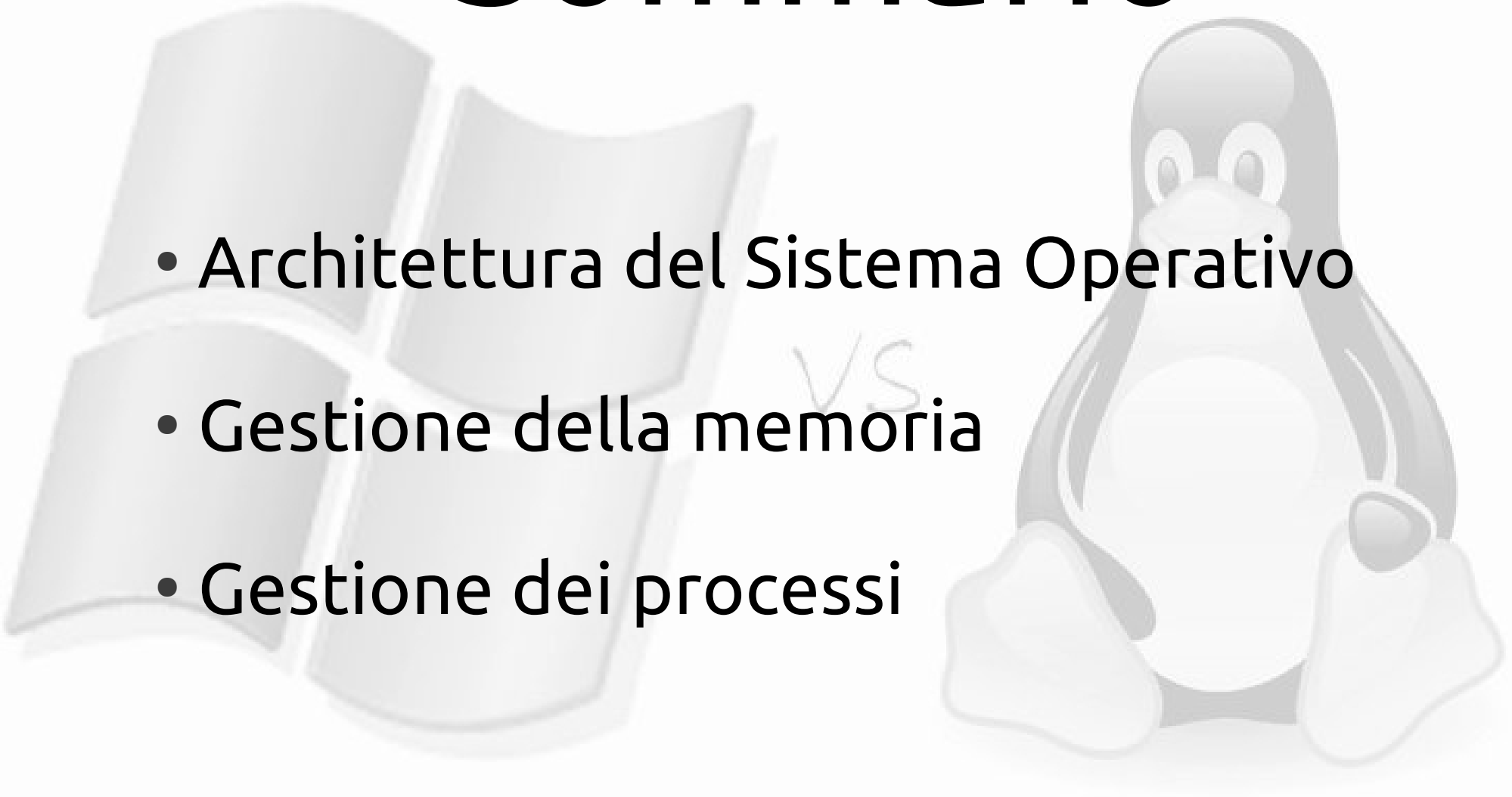


Syscall di Apache. Vi sembra complicato...?

Syscall di Microsoft IIS.

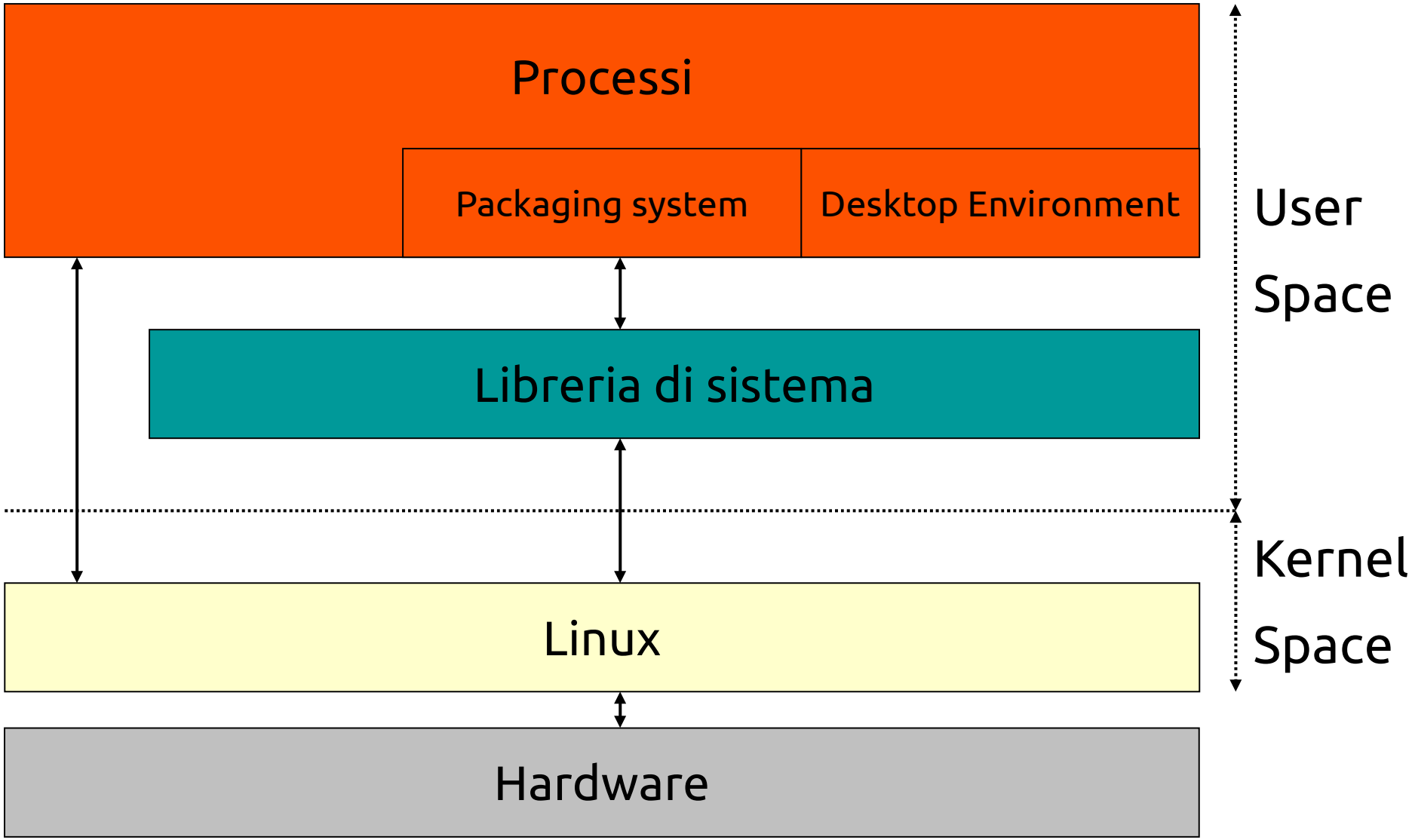


Sommario

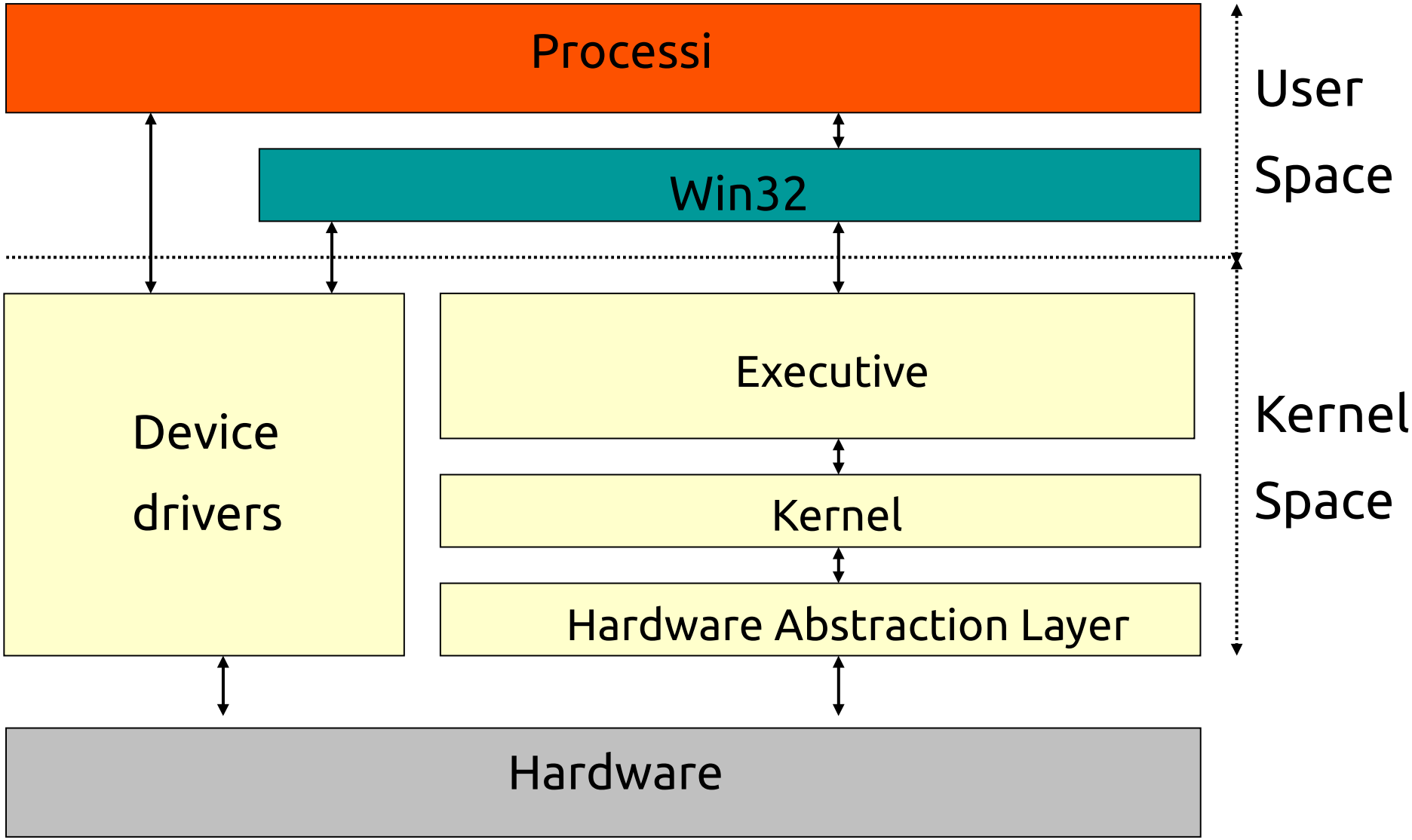


- Architettura del Sistema Operativo
- Gestione della memoria
- Gestione dei processi

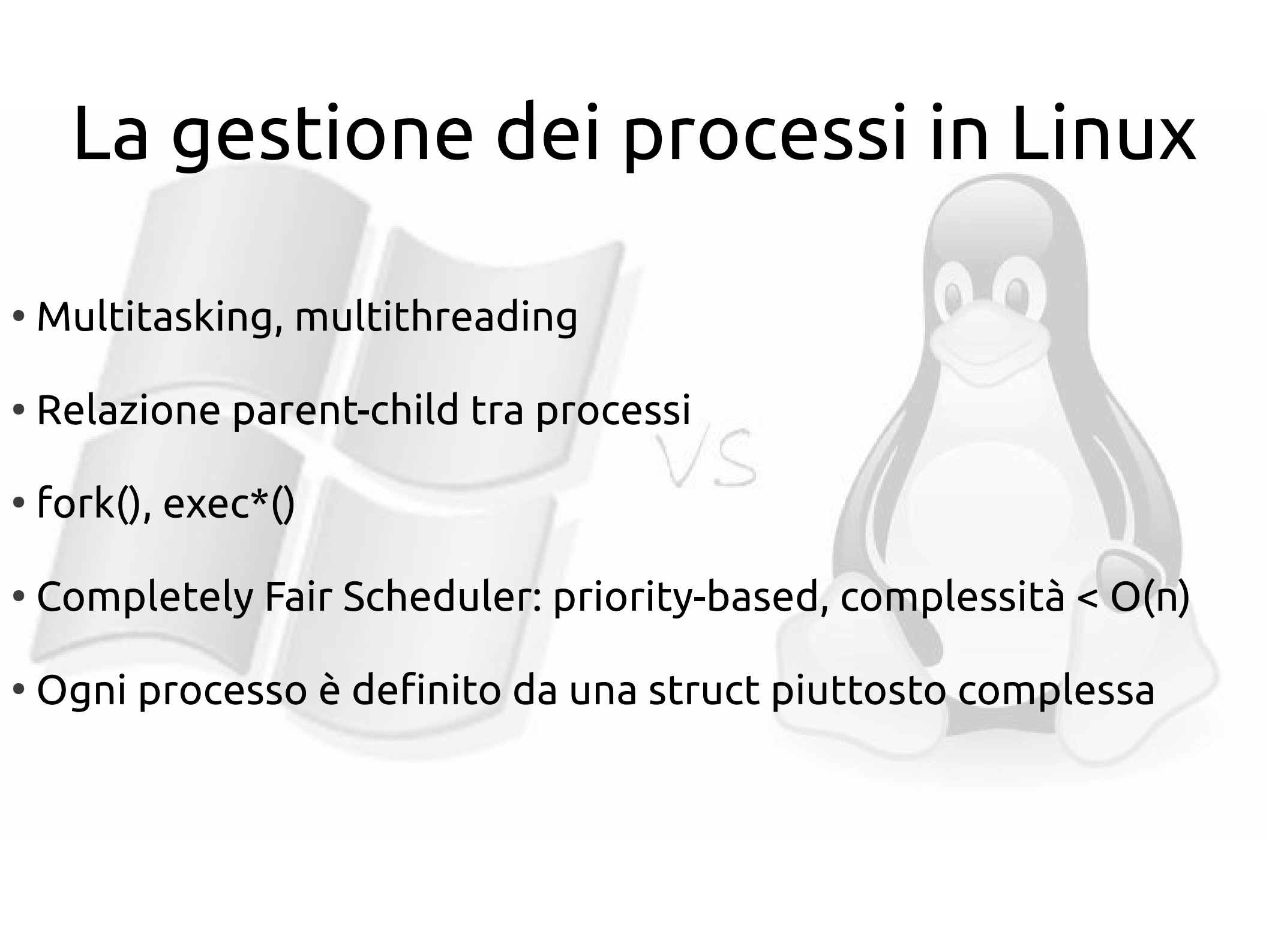
Architettura di GNU/Linux



Architettura di Windows



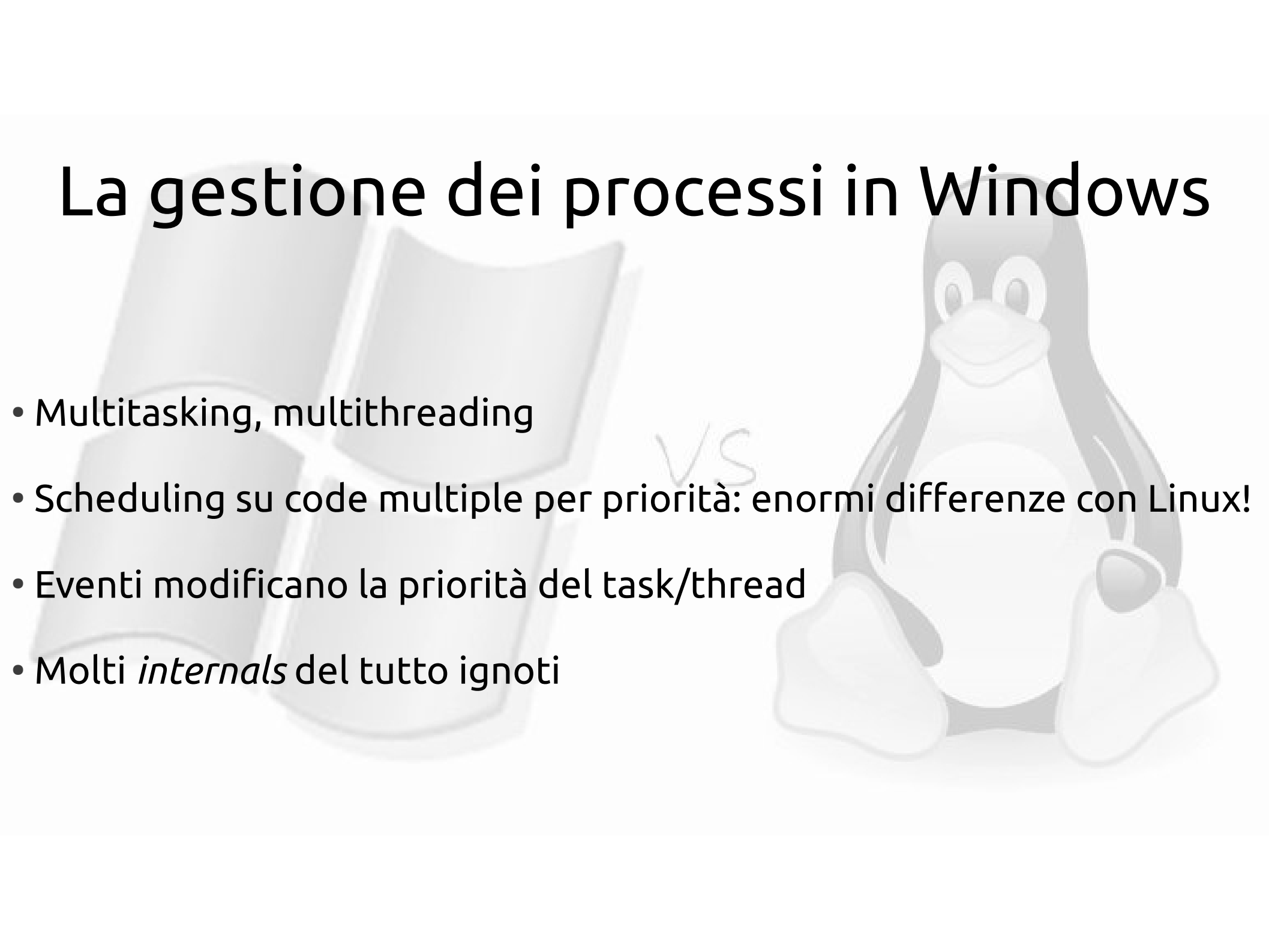
La gestione dei processi in Linux



- Multitasking, multithreading
- Relazione parent-child tra processi
- fork(), exec*()
- Completely Fair Scheduler: priority-based, complessità < O(n)
- Ogni processo è definito da una struct piuttosto complessa

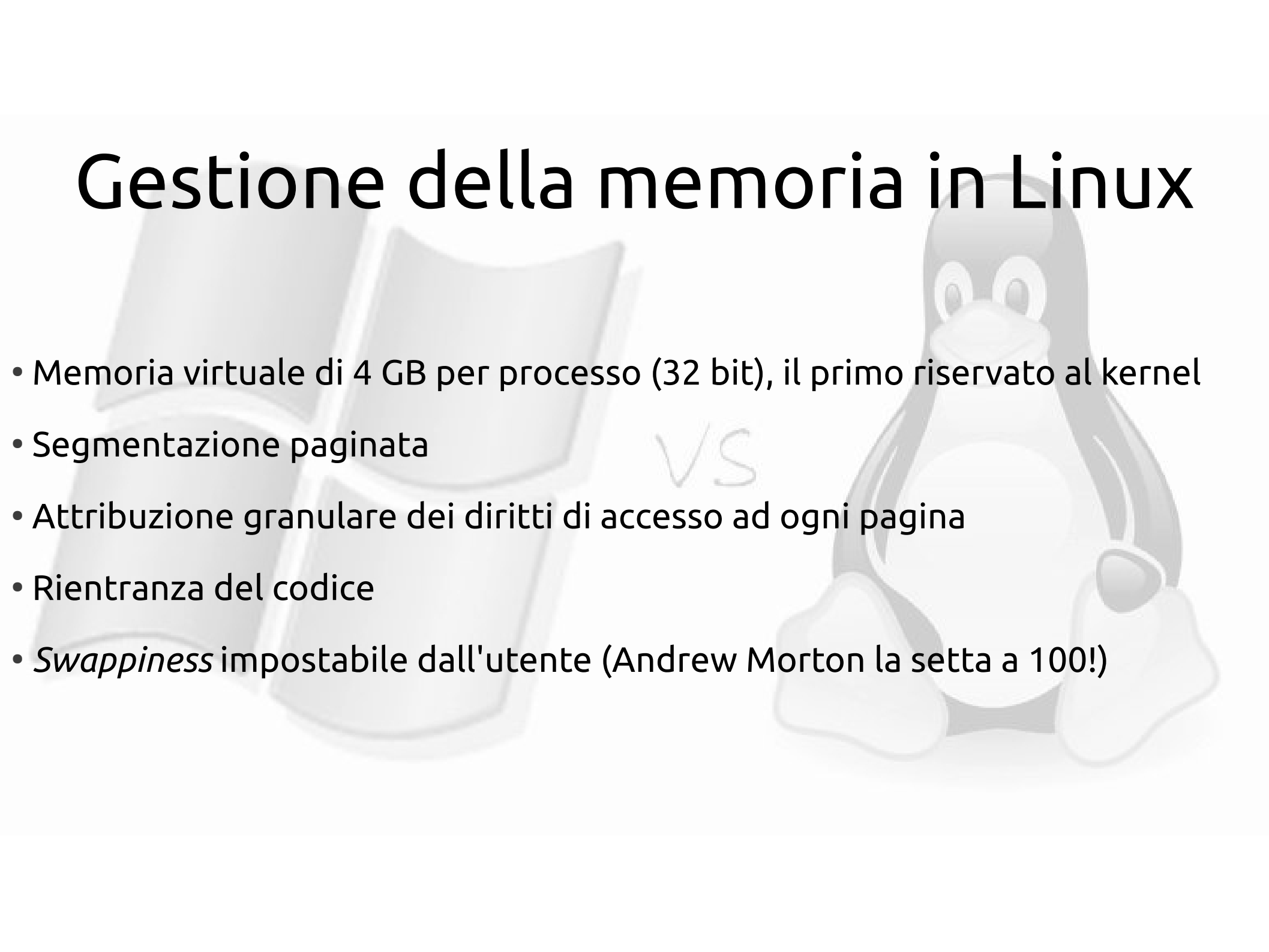
```
1
2  struct task_struct {
3          volatile long state;    /* -1 unrunnable, 0 runnable, >0 stopped */
4          void *stack;
5          unsigned int flags;     /* per process flags, defined below */
6
7          int prio, static_prio, normal_prio;
8          unsigned int rt_priority;
9          const struct sched_class *sched_class;
10         struct sched_entity se;
11         struct sched_rt_entity rt;
12
13         cpumask_t cpus_allowed;
14
15 #if defined(CONFIG_SCHEDSTATS) || defined(CONFIG_TASK_DELAY_ACCT)
16         struct sched_info sched_info;
17 #endif
18
19         struct mm_struct *mm, *active_mm;
20 /* task state */
21         int exit_state;
22         int exit_code, exit_signal;
23         int pdeath_signal;  /*  The signal sent when the parent dies  */
24
25         unsigned did_exec:1;
26         unsigned in_execve:1;   /* Tell the LSMs that the process is doing an
27                                  * execve */
28         unsigned in_iowait:1;
29
30         pid_t pid;
31         pid_t tgid;
32
33         struct task_struct *parent; /* recipient of SIGCHLD, wait4() reports */
34         struct list_head children;      /* list of my children */
35         struct list_head sibling;       /* linkage in my parent's children list */
36         struct task_struct *group_leader;       /* threadgroup leader */
37
38         struct list_head ptraced;
39         struct list_head ptrace_entry;
40
41         /* PID/PID hash table linkage. */
42         struct pid_link pids[PIDTYPE_MAX];
43         struct list_head thread_group;
44
45         struct completion *vfork_done;          /* for vfork() */
46         int __user *set_child_tid;              /* CLONE_CHILD_SETTID */
47         int __user *clear_child_tid;            /* CLONE_CHILD_CLEARTID */
48
49         cputime_t utime, stime, utimescaled, stimescaled;
50         cputime_t gtime;
51 #ifndef CONFIG_VIRT_CPU_ACCOUNTING
52         cputime_t prev_utime, prev_stime;
53 #endif
```

La gestione dei processi in Windows



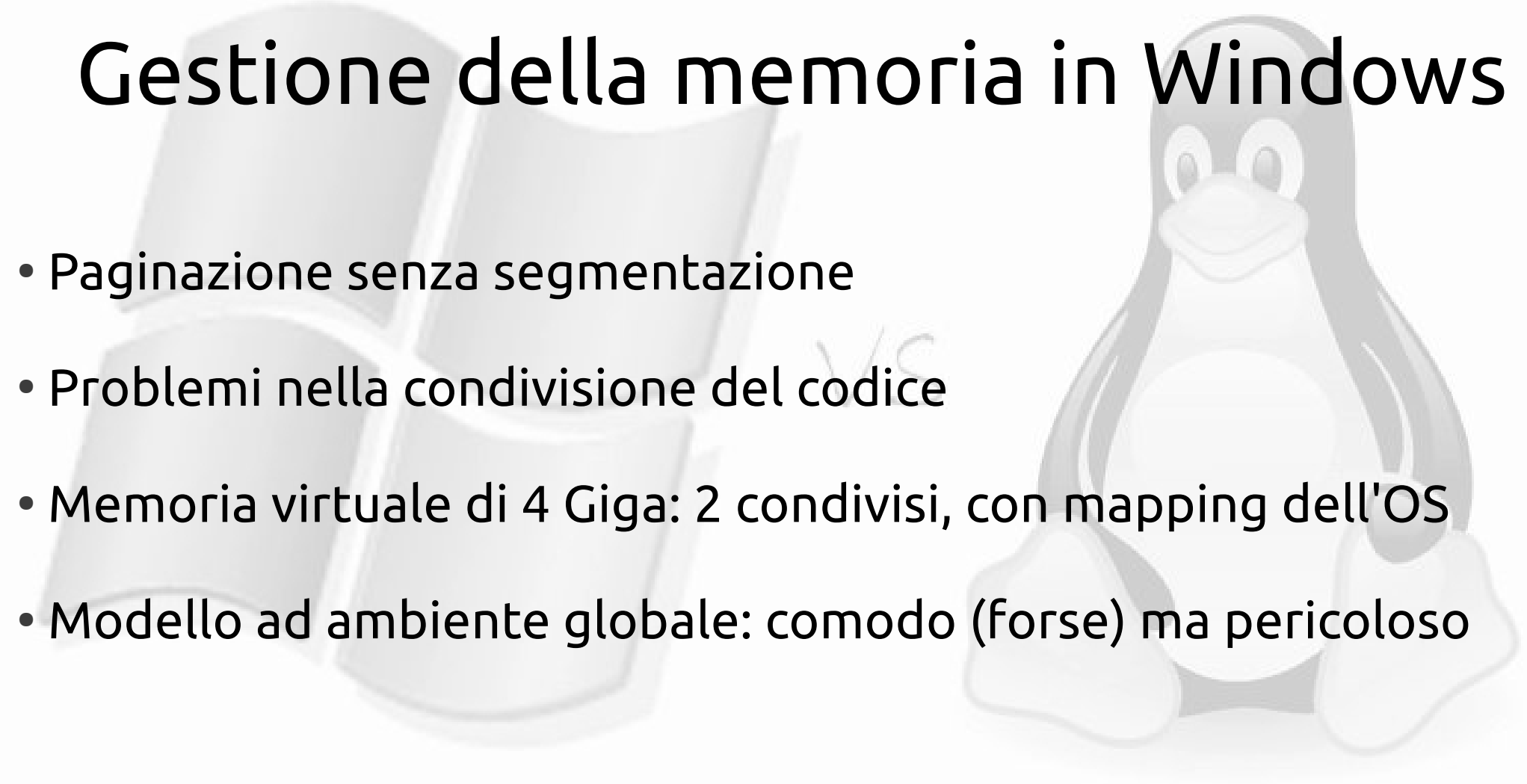
- Multitasking, multithreading
- Scheduling su code multiple per priorità: enormi differenze con Linux!
- Eventi modificano la priorità del task/thread
- Molti *internals* del tutto ignoti

Gestione della memoria in Linux



- Memoria virtuale di 4 GB per processo (32 bit), il primo riservato al kernel
- Segmentazione paginata
- Attribuzione granulare dei diritti di accesso ad ogni pagina
- Rientranza del codice
- *Swappiness* impostabile dall'utente (Andrew Morton la setta a 100!)

Gestione della memoria in Windows



- Paginazione senza segmentazione
- Problemi nella condivisione del codice
- Memoria virtuale di 4 Giga: 2 condivisi, con mapping dell'OS
- Modello ad ambiente globale: comodo (forse) ma pericoloso

And the winner is...?

Suggerimento: non comincia per “W”! :)


First they ignore you,
then they laugh at you,
then they fight you,
then you win.
-ghandi
Linux

Andrea Colangelo
warp10@ubuntu.com

@andreacolangelo
facebook.com/andrea.colangelo