



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

Sistemi Operativi¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2015/16

¹ © 2008–16 M. Monga. Creative Commons Attribuzione — Condividi allo stesso modo 4.0 Internazionale. <http://creativecommons.org/licenses/by-sa/4.0/deed.it>. Immagini tratte da [2] e da Wikipedia.



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

Lezione II: Introduzione laboratorio



- 6 (Bruschi) + 4 (Monga, Re) ore di lezione settimanali (12 crediti)
- Lezioni di teoria e in laboratorio
- Esame:
 - Scritto con domande a risposta multipla + orale
 - Prova pratica per la parte di laboratorio
- Libro di testo: Remzi e Andrea Arpaci-Dusseau *Operating Systems — Three easy pieces*
Versione online: <http://ostep.org>
- <http://homes.di.unimi.it/sisop/>
- <https://mameli.docenti.di.unimi.it/solab>

Things A Computer Scientist Rarely Talks About



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

“When I talk about computer science as a possible basis for insights about God, of course I’m not thinking about God as a super-smart intellect surrounded by large clusters of ultrafast Linux workstations and great search engines. That’s the user’s point of view.” [Donald E. Knuth]



```
Terminal
Terminal
welcome to god.creator GNU/Linux (1686, Kernel 2.6.x) on tty1
Mon Jan 01 00:00:00:01

login as: root
using keyboard-interactive authentication.
Password:
Last login: Mon Jan 01 00:00:00 0000 from 100.100.100.1
You have new mail in /var/spool/mail/god
Have a lot of fun...
god:~ # cd humansoul-lib-1.4beta6
god:~ # ./configure --prefix=/usr/local --with-pain=/usr/local --includedir
god:~ # make
god:~ # make install
```

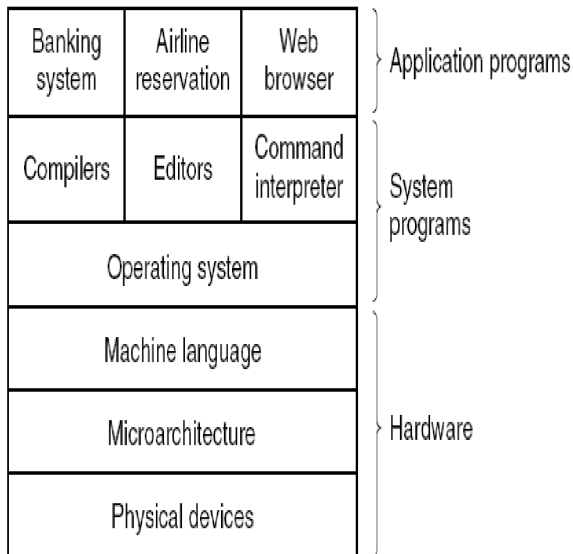


Cos'è un sistema operativo

Un insieme di programmi che:

- Gestisce in modo ottimale le risorse di un calcolatore;
- Facilita a programmatori e utenti finali l'uso della sottostante macchina hardware

The onion model



Sistemi Operativi

Bruschi
Monga Re

Concetti generali

La macchina fisica

Hardware
Concetti di base
Perché un s.o.

The onion model



Sistemi Operativi

Bruschi
Monga Re

Concetti generali

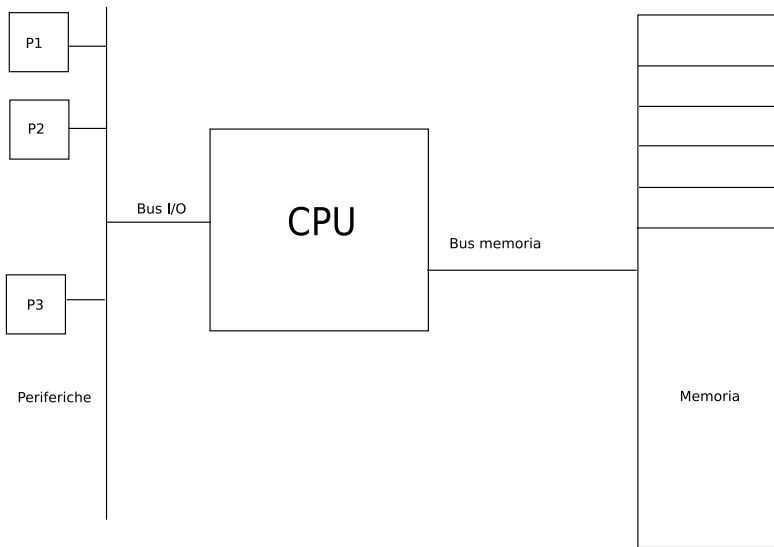
La macchina fisica

Hardware
Concetti di base
Perché un s.o.



- Il s.o. è l'unico programma che esegue con il totale controllo delle risorse hardware (**kernel mode**).
- Gli altri programmi si appoggiano unicamente sui servizi del s.o. e la loro esecuzione è gestita e controllata dal s.o. (**user mode**)
- In molti processori questa separazione è imposta via hardware

La macchina di Von Neumann



Sistemi Operativi

Bruschi Monga Re

Concetti generali

La macchina fisica

Hardware
Concetti di base
Perché un s.o.



- Registri a 32 bit
 - EAX, EBX, ECX, EDX,
 - ESI, EDI,
 - EBP, ESP,
 - EIP, EFLAGS
- Registri a 16 bit:
 - CS, DS, SS,
 - ES, FS, GS
- Real e Protected mode



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

- Si possono indirizzare direttamente porzioni di 8 bit, 1 byte ($AX = AH+AL$, $EAX = 16bit+AX$)
- Programmable Interrupt Controller (PIC): i8259 compatibile



I processori moderni hanno modalità di funzionamento in cui sono permesse operazioni diverse (*ring*), p.es. indirizzare tutta la memoria. i386 permette 4 ring diversi, di cui normalmente vengono usati solo 2 (Minix ne usa 3):

- 1 kernel (supervisor) mode
- 2 user mode



	Real mode	32-bit Protected mode
Protezioni hw	no	sí
Spazio di indirizzamento	2^{20}	2^{32}

- Real mode: memoria max 2^{20} byte, indirizzo ottenuto con due registri a 16 (SS:OFFSET)
*indirizzo = 16 * selettore + offset*
 - ci sono piú modi per riferirsi allo stesso indirizzo:
07C0:0000 e 0000:7C00 sono la stessa locazione fisica.
 - A20 gate
- Protected mode: il segmento è stabilito da un **descrittore** (che può essere cambiato solo in kernel mode)



- NASM, <http://nasm.sourceforge.org>
- PC Assembly Language, by Paul A. Carter
<http://www.drpaulcarter.com/pcasm/>
- Un altro assembler molto diffuso è gas
(<http://www.ibm.com/developerworks/linux/library/l-gas-nasm/index.html>)

```
1  mov eax, 3 ; eax = 3
2  mov bx, ax ; bx = ax
3  add eax, 4 ; eax = eax + 4
4  add al, ah ; al = al + ah
5  L8:db "A" ; *L8 = 'A'
6  mov al, [L8] ; al = *L8
```



Gli assembleri x86 si distinguono per la famiglia sintattica

Intel (nasm)	AT&T (as86, gas)
<code>mov ebx, eax</code>	<code>movl %eax, %ebx</code>
<code>mov eax, 42</code>	<code>movl \$42, %eax</code>
<code>mov [ebx], eax</code>	<code>movl %eax, 0(%ebx)</code>
<code>mov [ebx+4], eax</code>	<code>movl %eax, 4(%ebx)</code>
<code>mov byte [ebx], al</code>	<code>movb %eax, 0(%ebx)</code>
<code>call eax</code>	<code>call *%eax</code>

Concetti
generali

La macchina
fisica

Hardware

Concetti di base
Perché un s.o.



Qemu <http://fabrice.bellard.free.fr/qemu> PC (x86 or x86_64 processor)

- i440FX host PCI bridge and PIIX3 PCI to ISA bridge
- Cirrus CLGD 5446 PCI VGA card
- PS/2 mouse and keyboard
- 2 PCI IDE interfaces with hard disk and CD-ROM support
- Floppy disk
- NE2000 PCI network adapters
- Serial ports
- PCI UHCI USB controller and a virtual USB hub.

Ogni periferica è dotata di un **controller**. Il controller avrà registri che conservano lo stato della periferica. Come accedere (leggere o scrivere) al contenuto dei registri?

- 1 Spazi di indirizzamento separati chiamati **port**. Vi si accede con istruzioni particolari:
 - **out** port, **eax**
 - **in** **eax**, port
- 2 Memory-mapped I/O, lo spazio di indirizzamento è unico
 - **mov** [address], **eax**
 - **mov** **eax**, [address]



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

Cosa succede quando si accende un PC?



Cosa succede quando si accende un PC?

- 1 Inizia l'esecuzione del programma contenuto nel firmware (BIOS)



Cosa succede quando si accende un PC?

- 1 Inizia l'esecuzione del programma contenuto nel firmware (BIOS)
- 2 Il BIOS carica il programma contenuto nel **boot sector**



Cosa succede quando si accende un PC?

- 1 Inizia l'esecuzione del programma contenuto nel firmware (BIOS)
- 2 Il BIOS carica il programma contenuto nel **boot sector**
- 3 Il programma di boot carica il sistema operativo



Cosa succede quando si accende un PC?

- 1 Inizia l'esecuzione del programma contenuto nel firmware (BIOS)
- 2 Il BIOS carica il programma contenuto nel **boot sector**
- 3 Il programma di boot carica il sistema operativo
- 4 A questo punto il controllo della macchina è affidato al s.o., a cui dovranno essere richiesti i caricamenti di altri programmi

Programming the iron



Sistemi Operativi

Bruschi Monga Re

Concetti generali

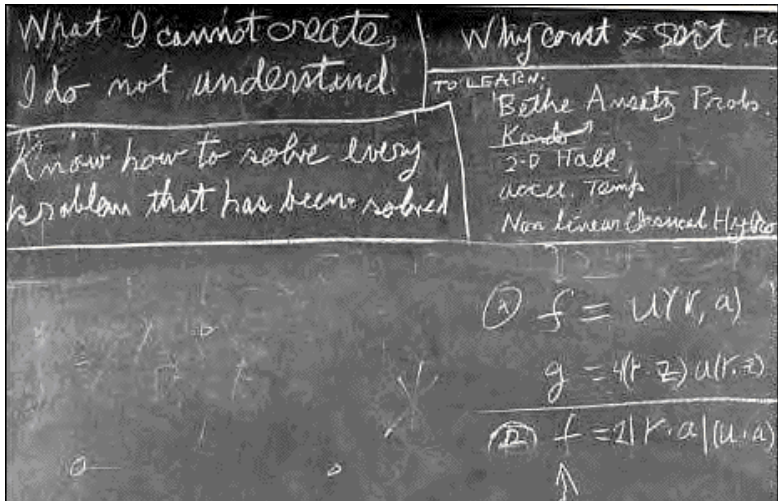
La macchina fisica

Hardware

Concetti di base

Perché un s.o.

What I cannot create I do not understand. [R. Feynman]





Programming the iron

```
1 bits 16 ; 16 bit real mode
2 org 0x7C00 ; origine indirizzo 0000:7C00
3
4 start:
5 mov ax, 0xb800 ; text video memory
6 mov ds, ax ; ds non accessibile direttamente
7 mov bx, 10
8 write:
9 cmp bx, 0
10 jz end
11 mov byte [ds:bx], 'm' ; indirizzamento relativo a ds
12 mov byte [ds:bx+1], 0x0F ; attrib = white on black
13 sub bx, 2
14 jmp write
15 end:
16 hlt
17
18 times 510-($-$$) db 0 ; 0-padding
19 dw 0xAA55
```

Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.



Programming the iron (2)

Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

```
1 start:
2     mov ax, 0xb800 ; text video memory dest array index [ES:DI]
3     mov es, ax
4     mov di, 0 ; start of video memory
5     cld ; clears direction flag (index regs incremented)
6     mov ah, 0x0F ; attrib = white on black
7     mov si, msg
8     call message
9     hlt
10
11 message:
12     lodsb ; carica un byte da [DS:SI] in AL e inc SI
13     cmp al, 0
14     jz done
15     stosw ; memorizza una word da [ES:DI] in AL e inc SI
16     jmp message
17 done: ret
18 msg db "Hello world from the bare machine!!!", 0
19 times 510-($-$$) db 0
20 dw 0xAA55
```

Programming the iron (3)



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.

```
1 waitenter: mov si, cont
2             call message
3             mov ah, 0
4             int 0x16 ; Wait for keypress (BIOS)
5             cmp al, 'm'
6             jz egg
7             cmp al, 'b'
8             jz basic
9             cmp al, 13
10            jnz waitenter
11            ret
12 egg: mov si, wow
13            call message
14            jmp waitenter
15 basic: int 0x18 ; basic (BIOS)
16            hlt
```



Sistemi
Operativi

Bruschi
Monga Re

Concetti
generali

La macchina
fisica

Hardware
Concetti di base
Perché un s.o.