



Sistemi
Operativi

Bruschi
Monga

JOS

Sistemi Operativi¹

Mattia Monga

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

a.a. 2012/13

¹  2011–13 M. Monga. Creative Commons Attribution-Condividi allo stesso modo 3.0 Italia License.
<http://creativecommons.org/licenses/by-sa/3.0/it/>. Immagini tratte da [?] e da Wikipedia.  1



Sistemi
Operativi

Bruschi
Monga

JOS

Lezione XVII: JOS



Servono 512MB di ram e `persistence-jos.qcow` in modo da salvare il proprio lavoro.

```
1 $ cd joslab
2 $ make
3 $ make qemu-nox
4
5 K> kerninfo
6 Special kernel symbols:
7   _start 0010000c (phys)
8   entry f010000c (virt) 0010000c (phys)
9   etext f0101a6d (virt) 00101a6d (phys)
10  edata f0112300 (virt) 00112300 (phys)
11  end f0112944 (virt) 00112944 (phys)
12 Kernel executable memory footprint: 75KB
```

Per uscire `Ctrl-a+x`



Sistemi
Operativi

Bruschi
Monga

JOS

Seguiremo

<http://pdos.csail.mit.edu/6.828/2012/labs/lab1/>

(spesso semplificando per motivi di tempo: non è vietato cercare di seguire tutti gli spunti del corso MIT! Tenete conto che gli studenti MIT hanno circa 2 settimane per realizzare gli obiettivi di ogni lab)

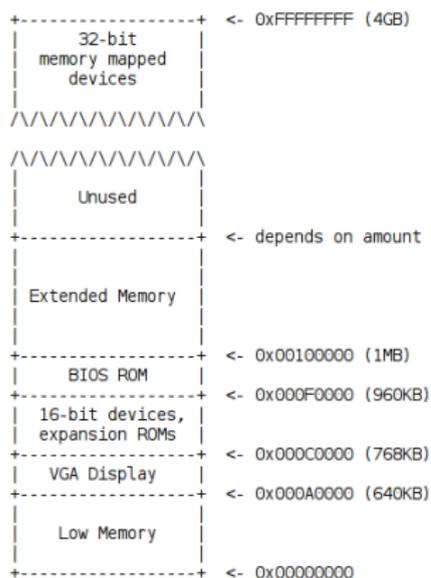
Layout della memoria



Sistemi Operativi

Bruschi Monga

JOS



La mappa della memoria è definita dal costruttore. Generalmente accessibile via firmware o con tecniche di probing (GRUB2 fornisce un comando lsmmmap)



Start

1 [f000:fff0] 0xffff0: ljmp \$0xf000,\$0xe05b

L'indirizzo fisico è calcolato secondo il Real-Mode addressing (a 16 bit)