



# Sistemi Operativi<sup>1</sup>

Mattia Monga

Dip. di Informatica e Comunicazione  
Università degli Studi di Milano, Italia

[mattia.monga@unimi.it](mailto:mattia.monga@unimi.it)

a.a. 2008/09



# Lezione XXXI: Gestione della memoria in MINIX



La gestione della memoria in MINIX è estremamente semplificata:

- Niente paginazione
- Niente swapping
- La memoria allocata ad un programma non può crescere, diminuire, né cambiare posizione assoluta

La gestione è responsabilità di PM (che è infatti talvolta chiamato anche memory manager MM)



Sono possibili due modelli:

- 1 Dati e Istruzioni combinati: text, data e stack sono un unico blocco (default)
- 2 Dati e Istruzioni separati: text è separato e può essere condiviso

# Memory layout

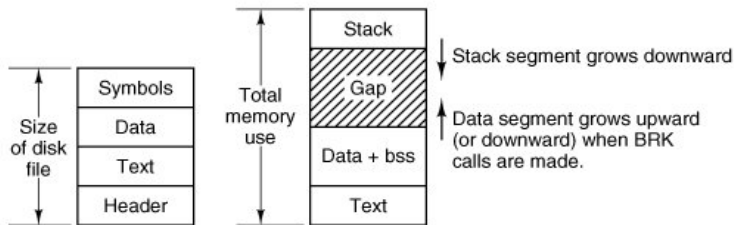


DICo

Sistemi Operativi

Bruschi  
Martignoni  
Monga

Memoria



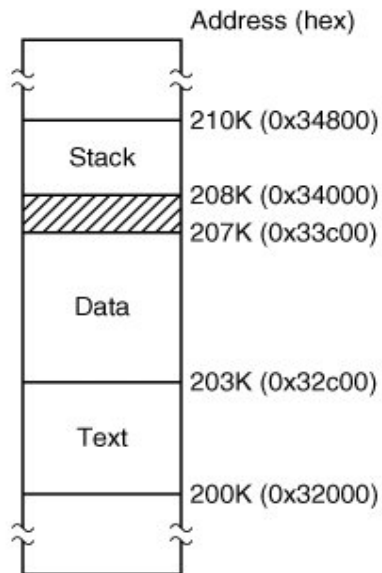
Le informazioni nello header possono essere cambiate col comando `chmem`



Le due strutture dati fondamentali sono:

- 1 la tabella dei processi `mproc` in `servers/pm/mproc.h`
- 2 la lista della memoria libera `hole` in `servers/pm/alloc.c`

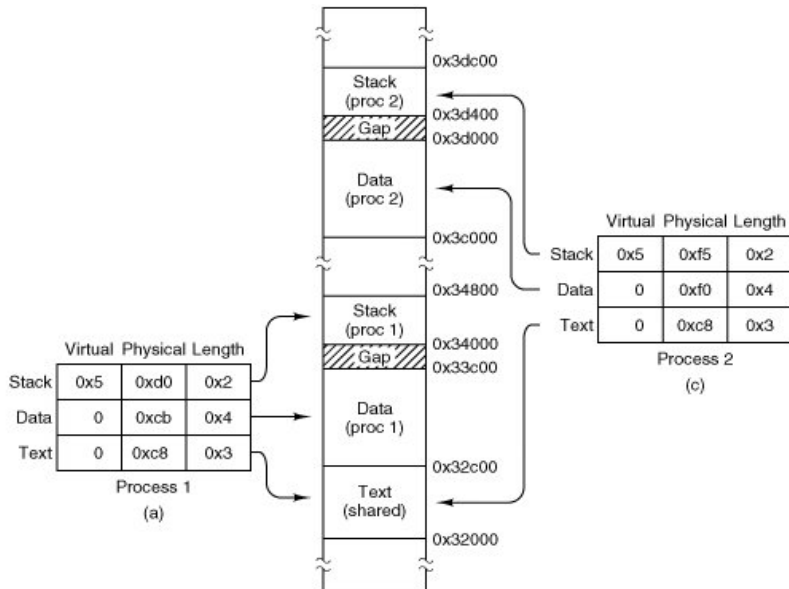
Il layout è definito dal campo `mp_seg`. Le misure sono sempre in `click`, che nel caso INTEL sono da 1024 byte.



	Virtual	Physical	Length
Stack	0x8	0xd0	0x2
Data	0	0xc8	0x7
Text	0	0xc8	0

(b)

	Virtual	Physical	Length
Stack	0x5	0xd0	0x2
Data	0	0xcb	0x4
Text	0	0xc8	0x3







```
1 /* servers/pm/alloc.c */
2 PRIVATE struct hole {
3     struct hole *h_next; /* pointer to next entry on the list */
4     phys_clicks h_base; /* where does the hole begin? */
5     phys_clicks h_len; /* how big is the hole? */
6 } hole[NR_HOLES];
```

# Esercizio 1



DICo

Sistemi  
Operativi

Bruschi  
Martignoni  
Monga

Memoria

Modificare l'allocazione di memoria in modo che avvenga con una strategia "best fit".

Visualizzare il livello di frammentazione esterna ottenuto prima e dopo il cambiamento.