

Process virtual memory

Sistemi Operativi A.A. 2019-2020

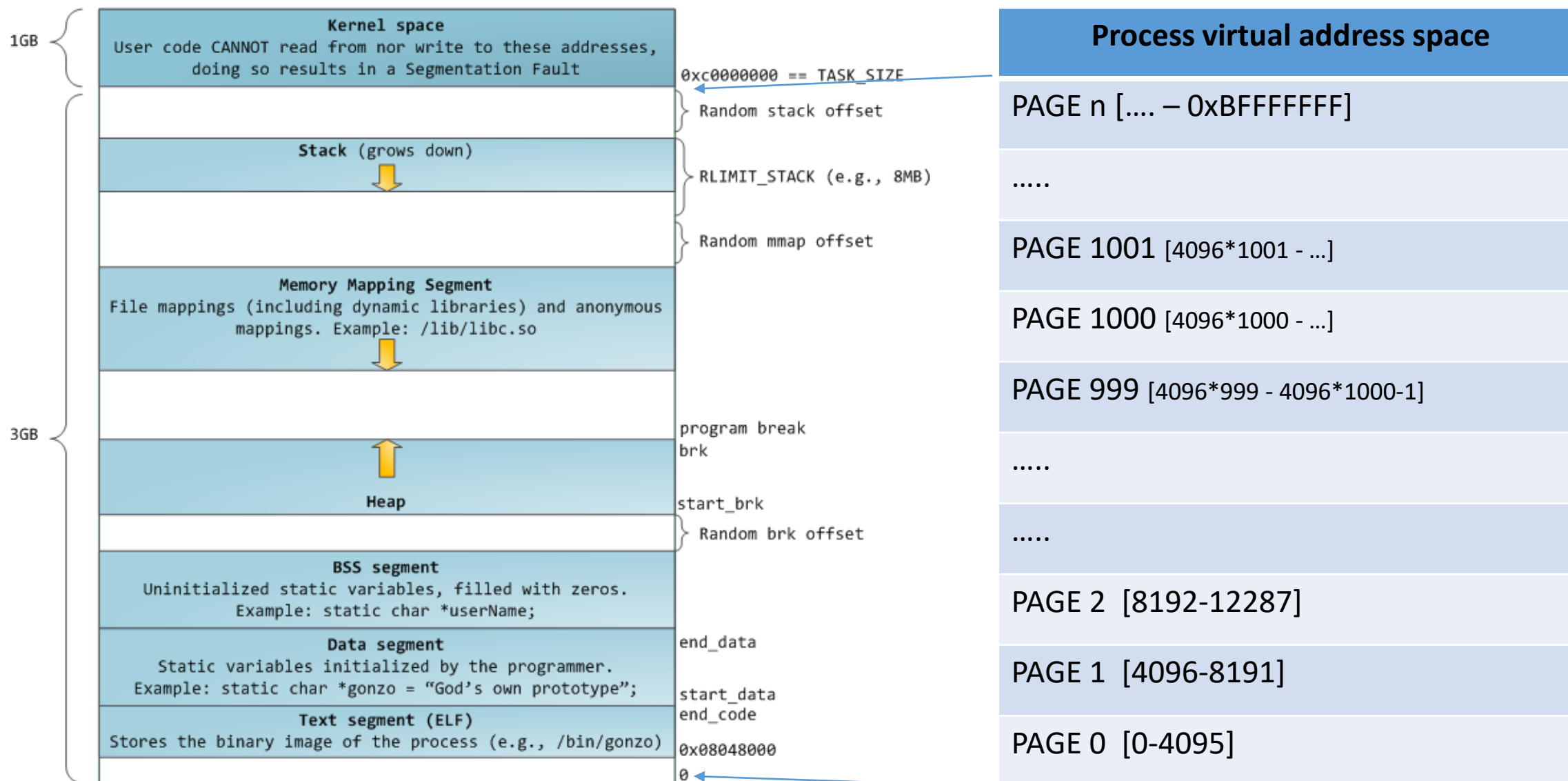
Marco Tessarotto

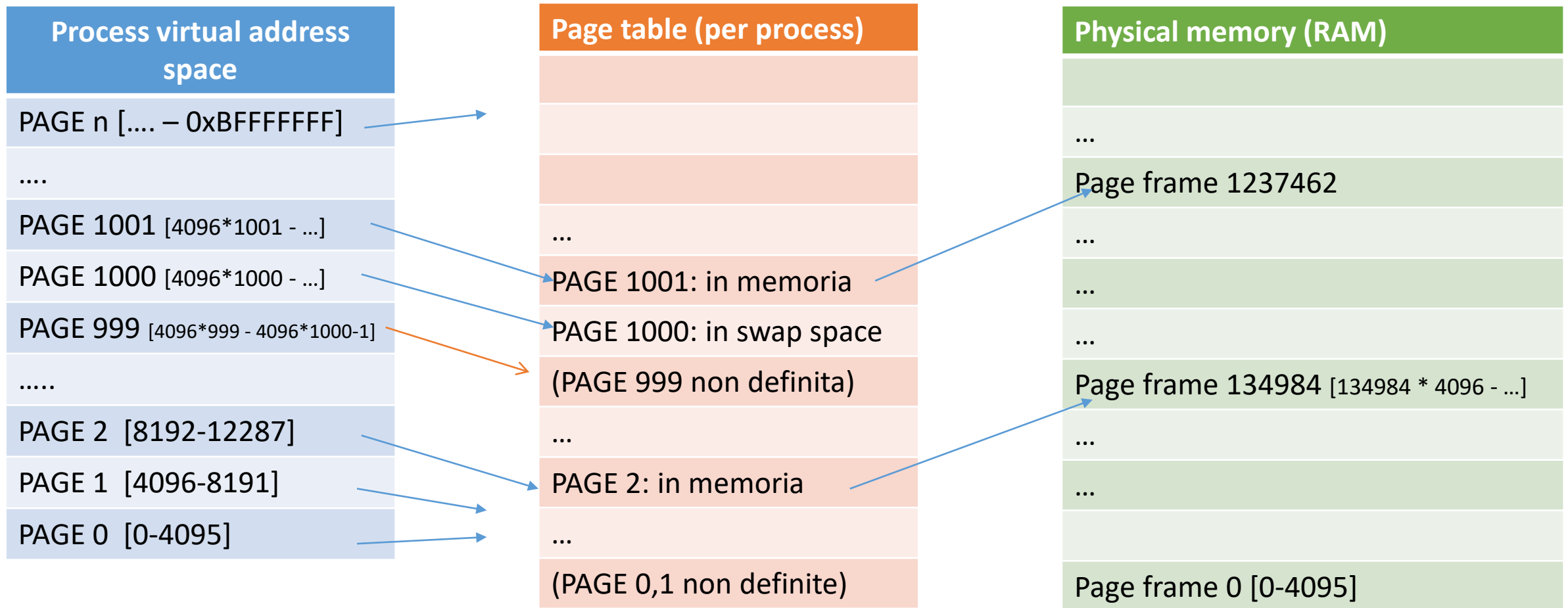
Gestione della memoria virtuale

- Fare riferimento al Kerrisk, sezioni 6.1, 6.2, 6.3, 6.4 (a partire da pag. 113)
- Il risultato della «**locality of reference**» (**principio di località**) è che è possibile eseguire un programma mantenendo solo una parte del suo spazio di indirizzi nella RAM.
- Una gestione della memoria virtuale divide la memoria utilizzata da ciascun processo in piccole unità di dimensioni fisse chiamate **pagine** (4096 bytes). Anche la RAM è divisa in una serie di «**page frame**» (cornici di pagina) della stessa dimensione.
- In qualsiasi momento, solo alcune delle pagine di un programma possono essere residenti in «page frame» di memoria fisica; queste pagine formano il cosiddetto «**resident set**» (insieme delle pagine residenti)
- Le copie delle pagine non utilizzate di un programma vengono mantenute nella **swap area** (area di scambio) — un'area riservata dello spazio su disco utilizzata per integrare la RAM del computer — e caricata nella memoria fisica solo se necessario.
- Quando un processo fa riferimento a una pagina che non è attualmente residente nella memoria fisica, si verifica un errore di pagina (**page fault**), a quel punto il kernel sospende l'esecuzione del processo mentre la pagina viene caricata dal disco nella memoria.

Gestione della memoria virtuale

- Memoria virtuale del processo: viene divisa in «**page**» (pagine), page size: 4096 byte
- Memoria fisica: divisa in «**page frame**» (cornici di pagina), page frame size: 4096 byte
- **Resident set**: insieme delle «page» (pagine) residenti in RAM
- **Virtual address space**: l'insieme di tutte le pagine rese disponibili al processo
- **Swap area** (area di scambio): un'area riservata dello spazio su disco utilizzata per integrare la (in aggiunta alla) RAM del computer; sono conservate le pagine non utilizzate dai processi
- Hardware + software (CPU + kernel) implementano la gestione della memoria virtuale





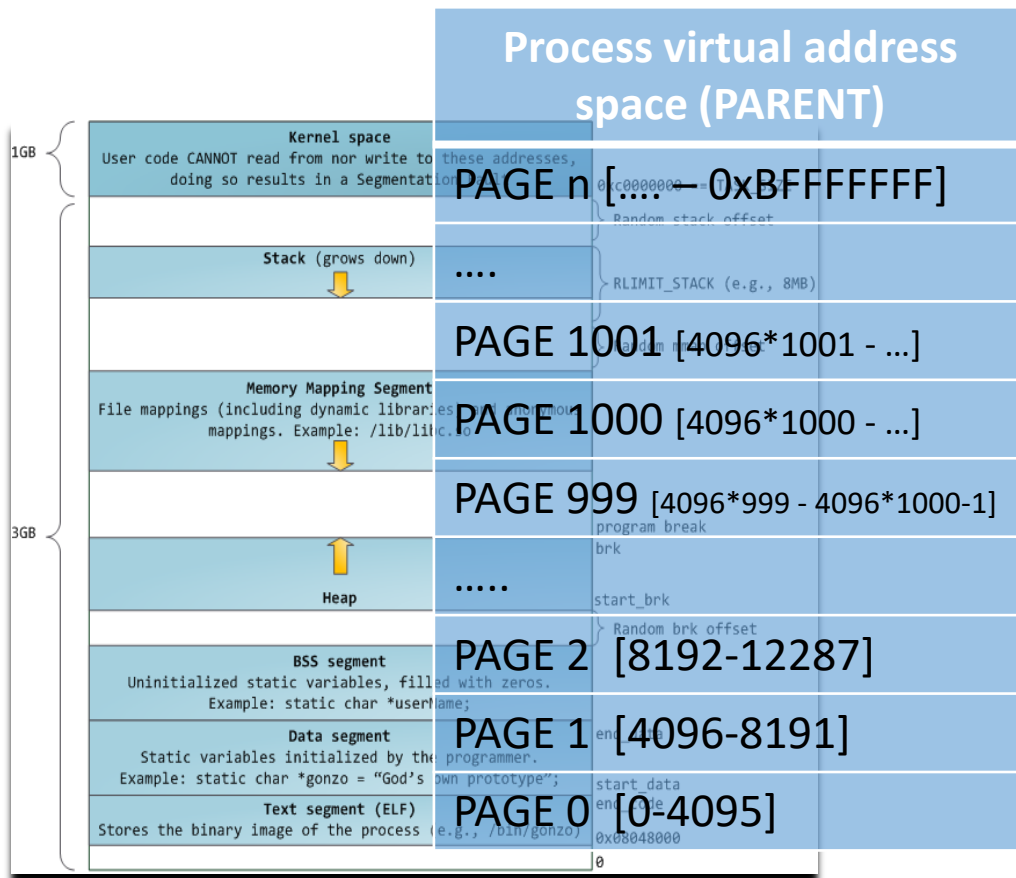
Indirizzo di memoria nel «virtual address space» del processo (shift right 12 bit) -> PAGE number

Lookup in process page table:

page non definita -> SIGSEGV signal

page ok e presente in memoria (parte del resident set) -> physical memory (page frame)

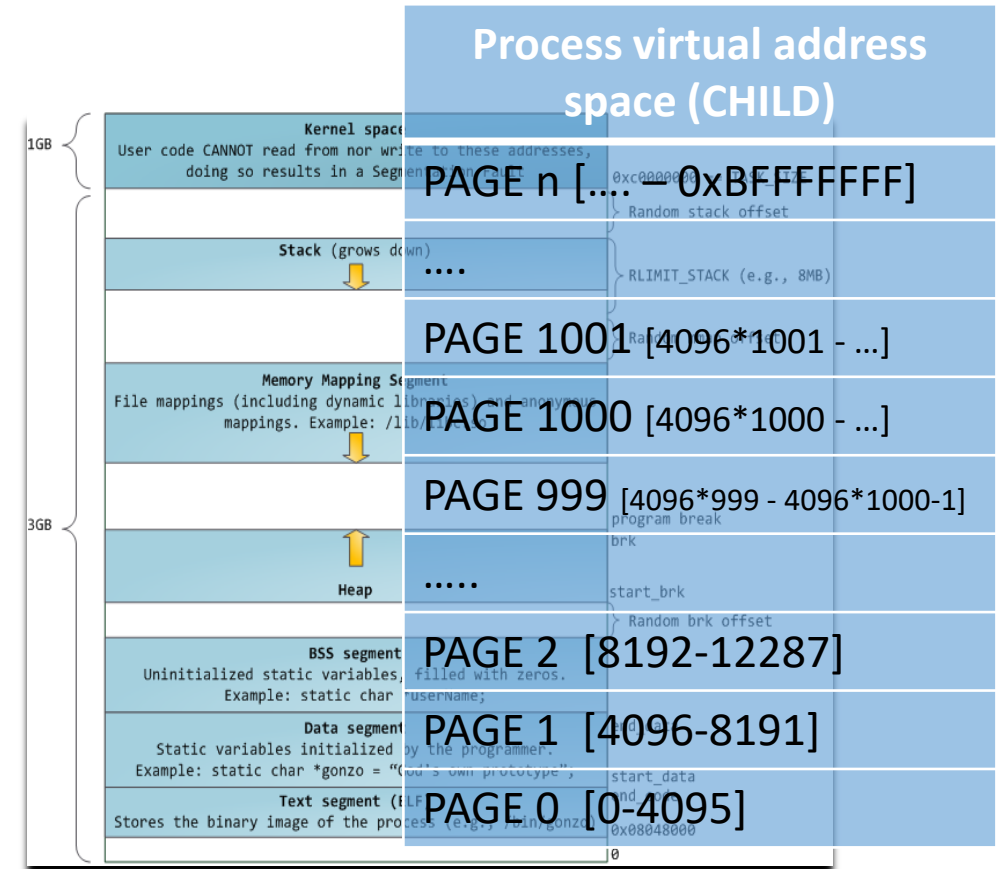
page ok e in swap space -> (kernel: trovare e/o fare spazio in physical mem. e) caricare da swap space la pagina



processo padre



fork()



processo figlio

system call fork() crea una COPIA identica della memoria virtuale del processo
QUINDI dopo fork gli indirizzi della memoria virtuale dei due processi sono IDENTICI ma distinti

processo padre

Process virtual address space
PAGE n [... - 0xBFFFFFFF]
....
PAGE 1001 [4096*1001 - ...]
PAGE 1000 [4096*1000 - ...]
PAGE 999 [4096*999 - 4096*1000-1]
.....
PAGE 2 [8192-12287]
PAGE 1 [4096-8191]
PAGE 0 [0-4095]

Page table (per process)
...
PAGE 1001: in memoria
PAGE 1000: in swap space
(PAGE 999 non definita)
...
PAGE 2: in memoria
...
(PAGE 0,1 non definite)

subito dopo fork(),
contenuto di page frame
3123498 == contenuto di
page frame 1237462 etc
etc

Physical memory (RAM)
...
...
Page frame 3123498 [...]
...
...
Page frame 1237462 [...]
...
...
Page frame 134984 [134984 * 4096 - ...]
...
...
Page frame 99454 [...]
...
...
Page frame 0 [0-4095]

processo figlio

Process virtual address space
PAGE n [... - 0xBFFFFFFF]
....
PAGE 1001 [4096*1001 - ...]
PAGE 1000 [4096*1000 - ...]
PAGE 999 [4096*999 - 4096*1000-1]
.....
PAGE 2 [8192-12287]
PAGE 1 [4096-8191]
PAGE 0 [0-4095]

Page table (per process)
...
PAGE 1001: in memoria
PAGE 1000: in swap space
(PAGE 999 non definita)
...
PAGE 2: in memoria
...
(PAGE 0,1 non definite)

Memory Mapping Segment
File mappings (including dynamic libraries) and anonymous mappings. Example: /lib/libc.so

Process virtual address space
PAGE n [... - 0xBFFFFFFF]
....
PAGE 1001 [4096*1001 - ...]
PAGE 1000 [4096*1000 - ...]
PAGE 999 [4096*999 - 4096*1000-1]
.....
PAGE 2 [8192-12287]
PAGE 1 [4096-8191]
PAGE 0 [0-4095]

Page table (per process)
...
PAGE 1001: in memoria
PAGE 1000: in swap space (PAGE 999 non definita)
...
PAGE 2: in memoria
...
(PAGE 0,1 non definite)

Physical memory (RAM)
...
...
...
...
...
...
...
Page frame 2545631 [...]
...
...
...
...
...
Page frame 0 [0-4095]

Memory Mapping Segment
File mappings (including dynamic libraries) and anonymous mappings. Example: /lib/libc.so

Process virtual address space
PAGE n [... - 0xBFFFFFFF]
....
PAGE 1001 [4096*1001 - ...]
PAGE 1000 [4096*1000 - ...]
PAGE 999 [4096*999 - 4096*1000-1]
.....
PAGE 2 [8192-12287]
PAGE 1 [4096-8191]
PAGE 0 [0-4095]

Page table (per process)
...
PAGE 1001: in memoria
PAGE 1000: in swap space (PAGE 999 non definita)
...
PAGE 2: in memoria
...
(PAGE 0,1 non definite)

memory map condivisa tra due processi (dopo fork())

Ipotesi: il memory map segment è contenuto tutto in una singola page (PAGE 1001)