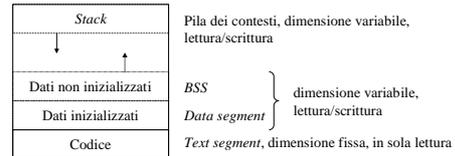


## Esame del Sistema Operativo Linux Parte 2 - Indice

1. Gestione della memoria
2. Gestione dell'I/O

## Esame del Sistema Operativo Linux Gestione della memoria - 1

- Massima semplicità per massima portabilità su diverse architetture fisiche
- Uno spazio di indirizzamento virtuale distinto per ogni processo, suddiviso in 4 sezioni



## Esame del Sistema Operativo Linux Gestione della memoria - 2

- Il segmento dati cresce o si riduce a seconda delle richieste del programma (p.es.: la `malloc()` di C)
  - POSIX non definisce queste chiamate di sistema
  - Una parte del segmento dati può ospitare *file* mappati in memoria
- Il segmento *stack* contiene l'ambiente (il contesto) d'esecuzione corrente e cresce nella direzione opposta del segmento dati
- Il segmento codice può essere condiviso tra più processi
  - Gli altri segmenti non lo sono tranne che per processi clonati da una `fork()`

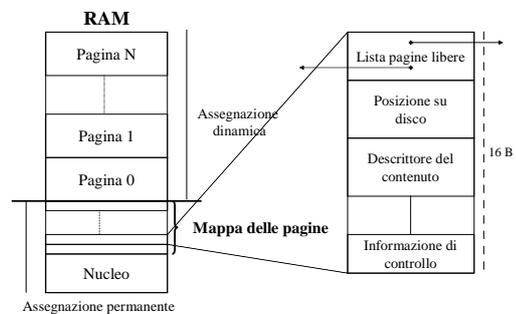
## Esame del Sistema Operativo Linux Gestione della memoria (UNIX) - 3

- In principio, l'allocazione della memoria principale avveniva mediante *swapping* di processi
  - Rimpiazzo di interi processi (loro spazio di indirizzamento) quando una particolare esecuzione rilevava mancanza di memoria
    - A seguito di `fork()`, di allocazione esplicita da programma, o di allocazione implicita da chiamata di procedura
  - Il gestore (*swapper*) creava lo spazio necessario salvando su disco i processi sospesi con più tempo d'esecuzione recente e minor priorità
    - Bastava utilizzare una lista dei blocchi liberi su disco

## Esame del Sistema Operativo Linux Gestione della memoria (UNIX) - 4

- Successivamente, fu introdotta paginazione con modalità "a richiesta" (*paging on demand*)
  - Un processo è eseguibile se il suo descrittore e la sua tabella delle pagine si trovano in RAM
    - Il suo spazio di indirizzamento è caricato da disco, per pagine, ad ogni riferimento che lo richiede
    - Nessun sforzo di anticipare il caricamento di pagine (*working set*)
  - Il processo 2 (*page daemon*) gestisce la situazione della pagine in RAM
    - Nucleo e mappa delle pagine (*core map*) restano sempre in RAM
    - Il resto è paginato con ciascuna pagina marcata secondo il proprio uso
      - Codice, dati, *stack*, tabella delle pagine (altrimenti in lista pagine libere)

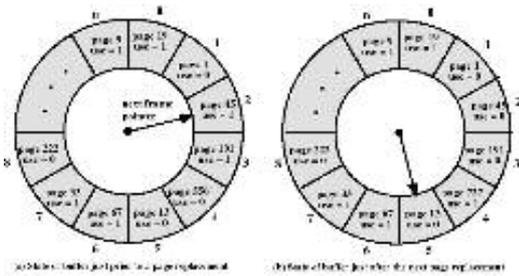
## Mappa delle pagine (core map)



Esame del Sistema Operativo Linux  
Gestione della memoria (UNIX) - 5

- *Page daemon* verifica periodicamente (1/4 s) che vi siano  $\geq \text{lotsfree}$  pagine libere
  - Se ne mancano, ne libera quante ne servono salvandone il contenuto su un'area di disco
  - La selezione delle pagine da rimpiazzare usa un algoritmo detto “della doppia passata” (*two-handed clock algorithm*)
    - Lista circolare delle pagine
    - La 1ª passata pone a 0 il *bit* di riferimento
    - La 2ª passata, a distanza data, rimuove le pagine nel frattempo non riferite (*bit* vale 1 altrimenti)

Clock algorithm (ad una passata)



Esame del Sistema Operativo Linux  
Gestione della memoria (UNIX) - 6

- *Page daemon* limita anche la frequenza di paginazione spostando alcuni processi su disco
  - Quelli che non abbiano eseguito negli ultimi 20 s
  - Tra i 4 più grandi, quello da più tempo in memoria
- Se vi è spazio libero, *page daemon* riporta in RAM processi pronti selezionati con una euristica di “valore”
  - Naturalmente caricando solo il descrittore di processo e la sua tabella delle pagine
    - Il resto viene caricato a seguito di esplicito riferimento

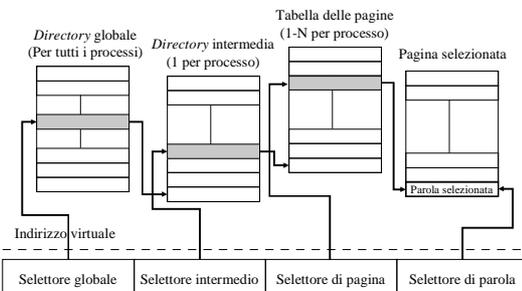
Esame del Sistema Operativo Linux  
Gestione della memoria (Linux) - 6

- Lo spazio di indirizzamento virtuale di processo è ampio 4 GB (per architetture a 32 bit)
  - 1 GB riservato (e non visibile in *modo normale*) per la sua tabella delle pagine ed altri dati di controllo ad uso del nucleo
- Spazio suddiviso in “regioni” costituite da sequenze contigue di pagine
  - Le regioni possono essere non consecutive
- Ciascuna regione ha un descrittore noto al nucleo
  - Lo spazio di indirizzamento virtuale di un processo è visto come una lista di descrittori

Esame del Sistema Operativo Linux  
Gestione della memoria (Linux) - 7

- In caso di `fork()` Linux replica per il figlio l'intera lista di descrittori del padre
- Le pagine sono copiate solo a seguito di un esplicito tentativo di modifica (*copy on write*)
  - La regione è marcata R/W, le sue pagine (dati) R
  - La richiesta di scrittura causa eccezione ed il nucleo copia la pagina per il richiedente, marcandola come R/W

Esame del Sistema Operativo Linux  
Gestione della memoria (Linux) - 8



### Esame del Sistema Operativo Linux Gestione della memoria (Linux) - 9

- Il nucleo rimane sempre in RAM
  - Di dimensione variabile a causa del caricamento dinamico di moduli di gestione dispositivi
- La RAM rimanente viene usata per
  - [P1] Le pagine attive dei processi utente
  - [P2] Una cache di blocchi di file usata dal file system
    - Di dimensione variabile e strutturata su pagine
  - [P3] Un insieme di pagine utente inattive ma presenti
- La RAM deve pertanto essere assegnata in porzioni di dimensione arbitraria

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 157

### Esame del Sistema Operativo Linux Gestione della memoria (Linux) - 10

- L'algoritmo di allocazione primario (*buddy algorithm*) usa il principio di prossimità
  - Ogni richiesta di ampiezza  $N$  è arrotondata a  $2^n \geq N$
  - La memoria disponibile viene frazionata in metà successive fino a frazioni di ampiezza  $2^n$ , di cui una va al richiedente
    - Una struttura ausiliaria contiene la testa di liste predefinite di frazioni di ampiezza  $2^i$  ( $i=0, \dots, n$ ) così da velocizzare la ricerca
  - La memoria disponibile usata per l'allocazione è sempre la frazione libera di minore dimensione
  - Al rilascio, ogni frazione tornata libera si unisce con la frazione vicina (*buddy*) se questa è libera
- Due algoritmi sussidiari cercano di ridurre la frammentazione causata dall'algoritmo

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 158

### Esame del Sistema Operativo Linux Gestione della memoria (Linux) - 11

- I segmenti codice ed i *file* mappati in memoria hanno un corrispondente *file* su disco
- Al resto (le aree di lavoro dei processi) viene assegnata una partizione paginata (*paging partition*) vista come un *byte stream* od un *file* di pagine su disco
  - La partizione paginata è d'uso più semplice e veloce
    - Nessun limite fissato di scrittura, scrittura contigua
- Le pagine libere sulla partizione e/o sul *file* sono indicate mediante *bitmap*
  - Tale spazio viene assegnato alle pagine di lavoro rimosse temporaneamente dalla RAM

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 159

### Esame del Sistema Operativo Linux Gestione della memoria (Linux) - 12

- Il *page daemon* di Linux, *kswapd*, ha periodo 1 s
  - Per ogni attivazione, esegue fino a 6 passate di un ciclo di lavoro, cercando pagine da spostare su disco
    - Tra quelle di tipo [P2] e [P3], con una variante del *clock algorithm*
    - Tra quelle condivise da più processi ma non molto usate
    - Tra quelle di tipo [P1]
      - Cominciando dal processo con più pagine, scandendo l'intera lista dei suoi descrittori di regione (per indirizzo virtuale) con *clock algorithm* ma solo sulle pagine attive
  - Ogni pagina selezionata modificata
    - Se ha corrispondente su disco viene posta in attesa di riscrittura
    - Altrimenti viene riportata sulla *paging partition* o sul *paging file*
- Un altro *daemon*, *bdflush*, si occupa della riscrittura

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 160

### Esame del Sistema Operativo Linux Gestione dell'I/O - 1

- La scelta originale di UNIX è stata trattare i dispositivi di I/O come *file* di tipo speciale, ciascuno con posizione specifica nel FS (*/dev/*)
  - *File* orientati a carattere (p.es.: tastiera, rete, ...)
  - *File* orientati a blocco (p.es.: disco)
- Un gestore (*device driver*) è associato in modo esclusivo a ciascun dispositivo (o famiglia di dispositivi dello stesso tipo)
  - Una coppia di indici <maggiore, minore> identifica precisamente ciascun dispositivo di I/O

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 161

### Esame del Sistema Operativo Linux Gestione dell'I/O - 2

- Linux consente il caricamento dinamico dei moduli necessari alla gestione di specifici dispositivi
  - Soluzione preferibile alla configurazione statica, che richiede la ricompilazione dell'intero nucleo, specie a fronte di grande varietà di *hardware*
- Il caricamento dinamico richiede varie azioni di configurazione
  - Rilocazione dello spazio di indirizzamento del modulo
  - Allocazione delle risorse necessarie (p.es.: interruzione)
  - Configurazione del vettore delle interruzioni
  - Attivazione ed inizializzazione del gestore

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 162

### Esame del Sistema Operativo Linux Gestione dell'I/O - 3

- Un particolare tipo di *file*, detto *socket*, modella la connessione di rete ed il protocollo ad essa associato
  - Può essere creato e distrutto dinamicamente
  - Ciascun *socket* è associato ad uno specifico indirizzo di rete
- 3 tipi di connessione con scelta alla creazione
  - Connessione affidabile a flusso di caratteri (p.es.: **TCP**)
    - Il S/O garantisce la correttezza della trasmissione
    - Invio e ricezione per blocchi di dimensione libera
  - Connessione affidabile a flusso di pacchetti
    - Come sopra, ma con invio e ricezione solo per pacchetti
  - Trasmissione inaffidabile di pacchetti
    - Dove l'utente si occupa di trattare gli eventuali errori

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 163

### Esame del Sistema Operativo Linux Gestione dell'I/O - 4

- Per velocizzare lettura e scrittura di dati su dispositivi a blocchi, una zona di RAM è usata come *cache* dedicata
  - Ogni blocco richiesto in lettura viene prima cercato in *cache*
  - Ogni blocco scritto viene trattenuto in *cache* il più a lungo possibile
    - Fino a quando la cache è piena e bisogna creare spazio
    - Fino all'attivazione del *daemon* di scrittura su disco

Esame del Sistema Operativo Linux Architettura degli elaboratori 2 - T. Vardanega Pagina 164