Struktur *Page Table*

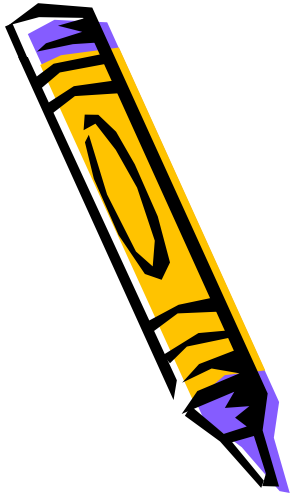
Heri Kurniawan

OS-Genap 2008/2009

Tujuan Pembelajaran

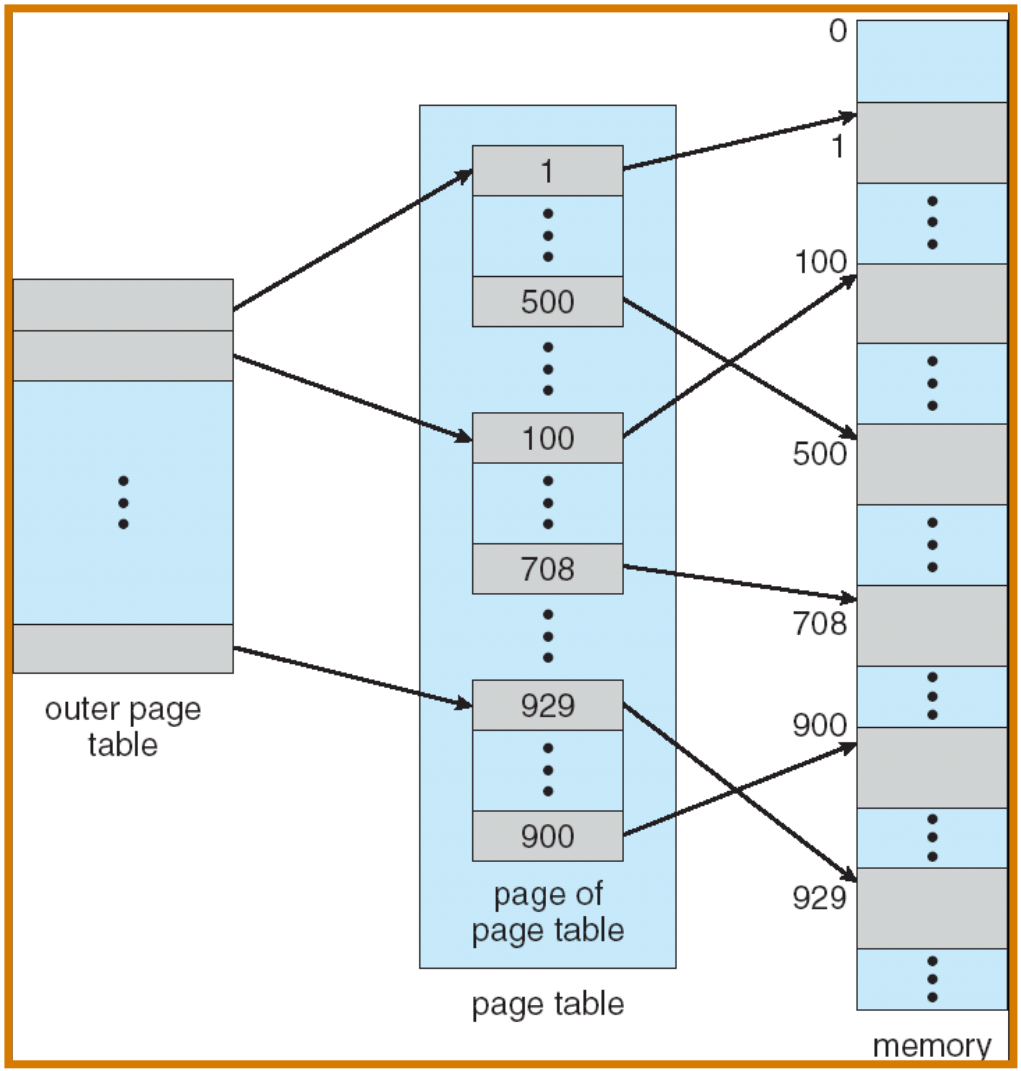
- Membahas teknik multi-level page table, hashed page table dan inverted page table

Hierarchical Page Tables



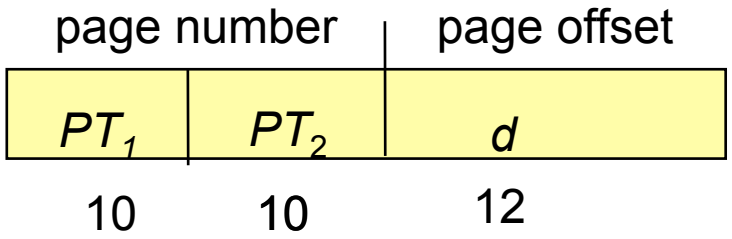
- Jika, Ruang alamat logika = 2^{32}, page size= 2^{12}, maka dibutuhkan page table sebanyak $2^{32}/2^{12}=2^{20}=1048576$ entry. Jika per entry=4bytes, maka dibutuhkan 4MB ruang untuk page table -> terlalu besar untuk dimuat ke memory.
- Solusinya ? Memecah page table menjadi lebih kecil

Skema Page-Table dua tingkat (Pentium)

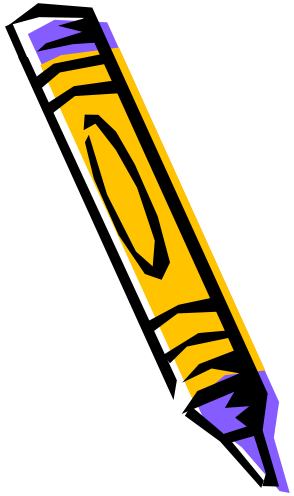


Page Table dua tingkat

- Ruang alamat logika = 2^{32},page size= 2^{12},
  - page number terdiri dari 20 bit
  - page offset terdiri dari 12 bit
- Karena page tabel juga dipaging, page number dibagi menjadi
  - 10 bit page number
  - 10 bit page offset
- PT_1=Outer Page; PT_2=Page of Page Table

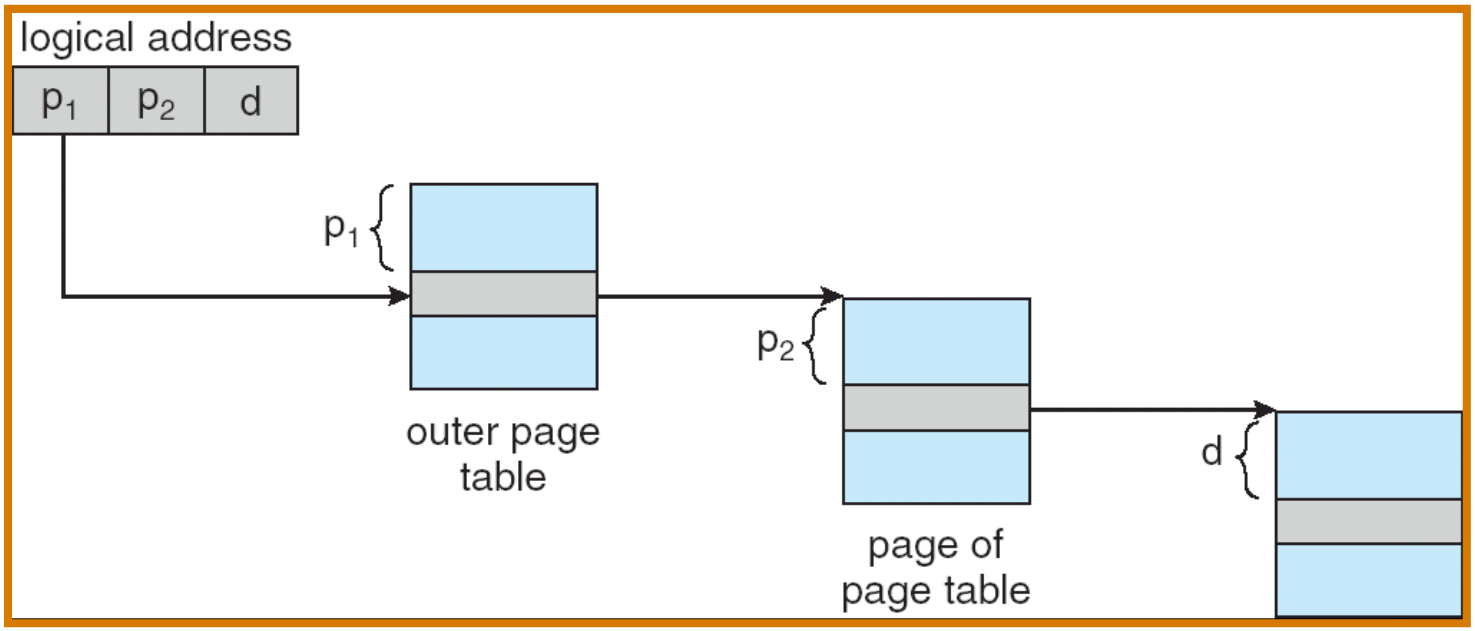


Struktur Page Table

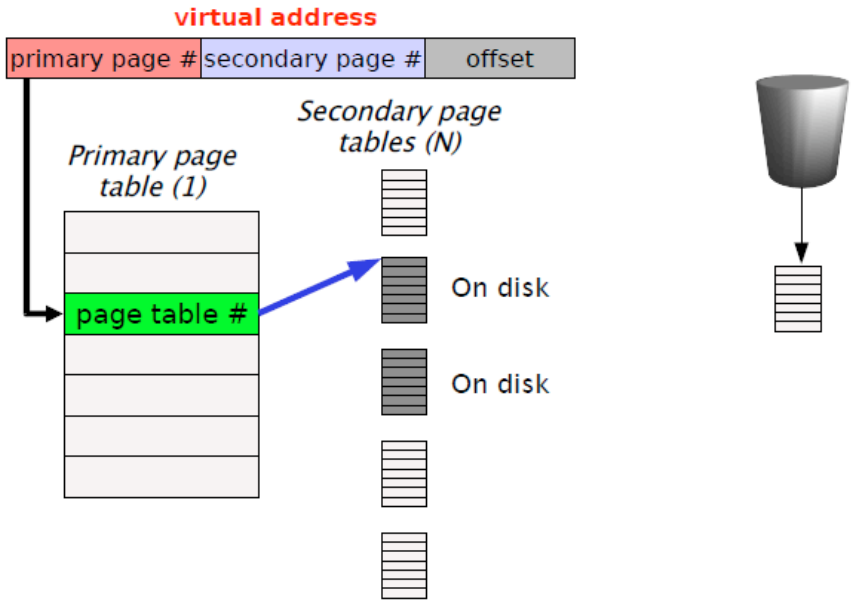


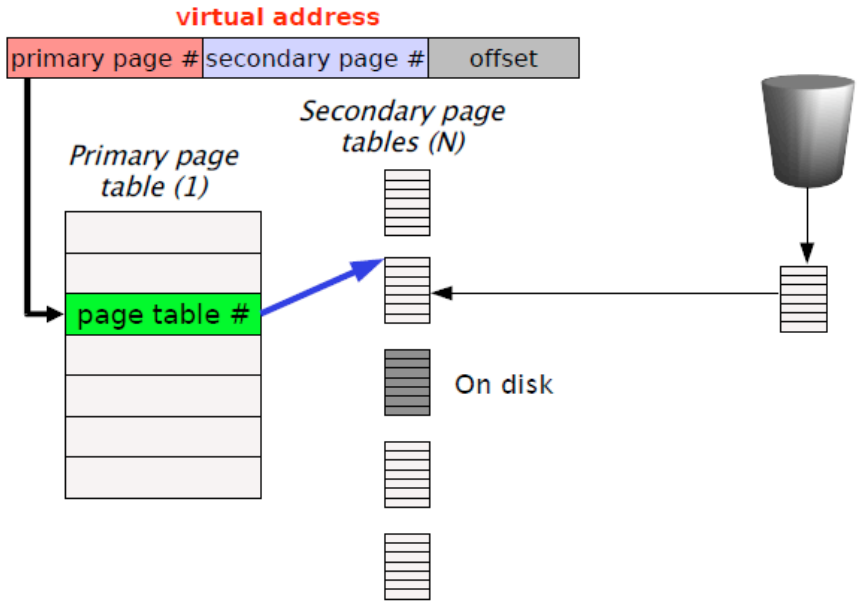
- Hierarchical Paging
  - Paging the page table
  - 2 Level (Pentium)
  - 3 Level (SPARC)
  - 4 Level (Motorola 68030)
  - Kurang baik untuk arsitektur 64 bit
- Hashed Page Tables
  - jika ruang alamat lebih dari 32 bit
- Inverted Page Tables
  - 64-bit UltraSPARC, Power PC, IA-64

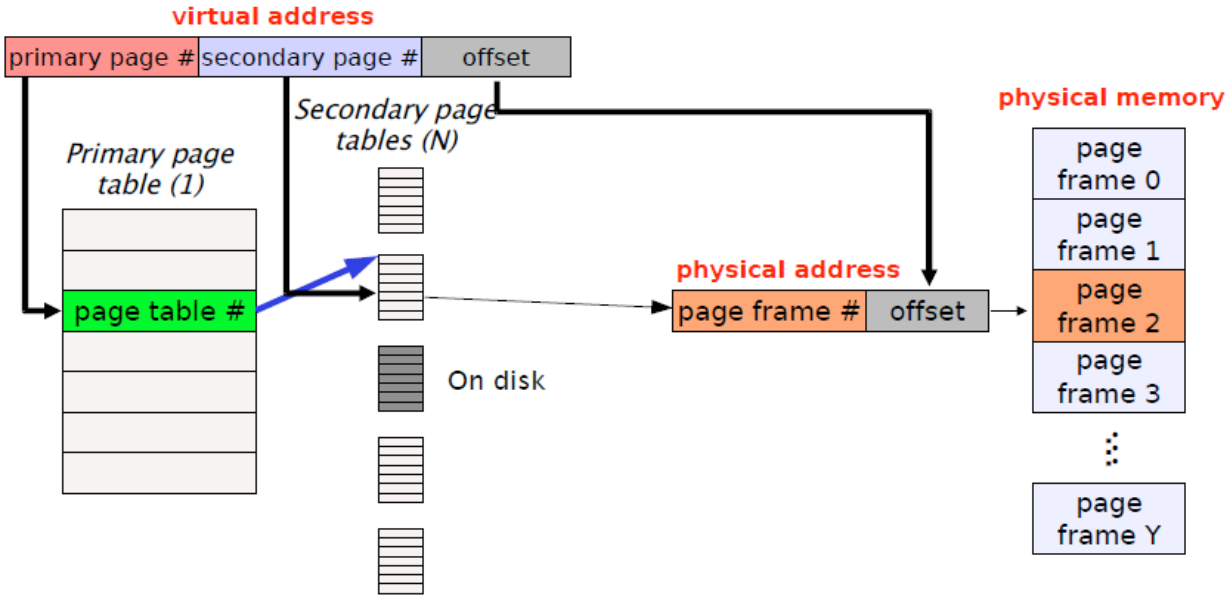
Skema penerjemahan alamat



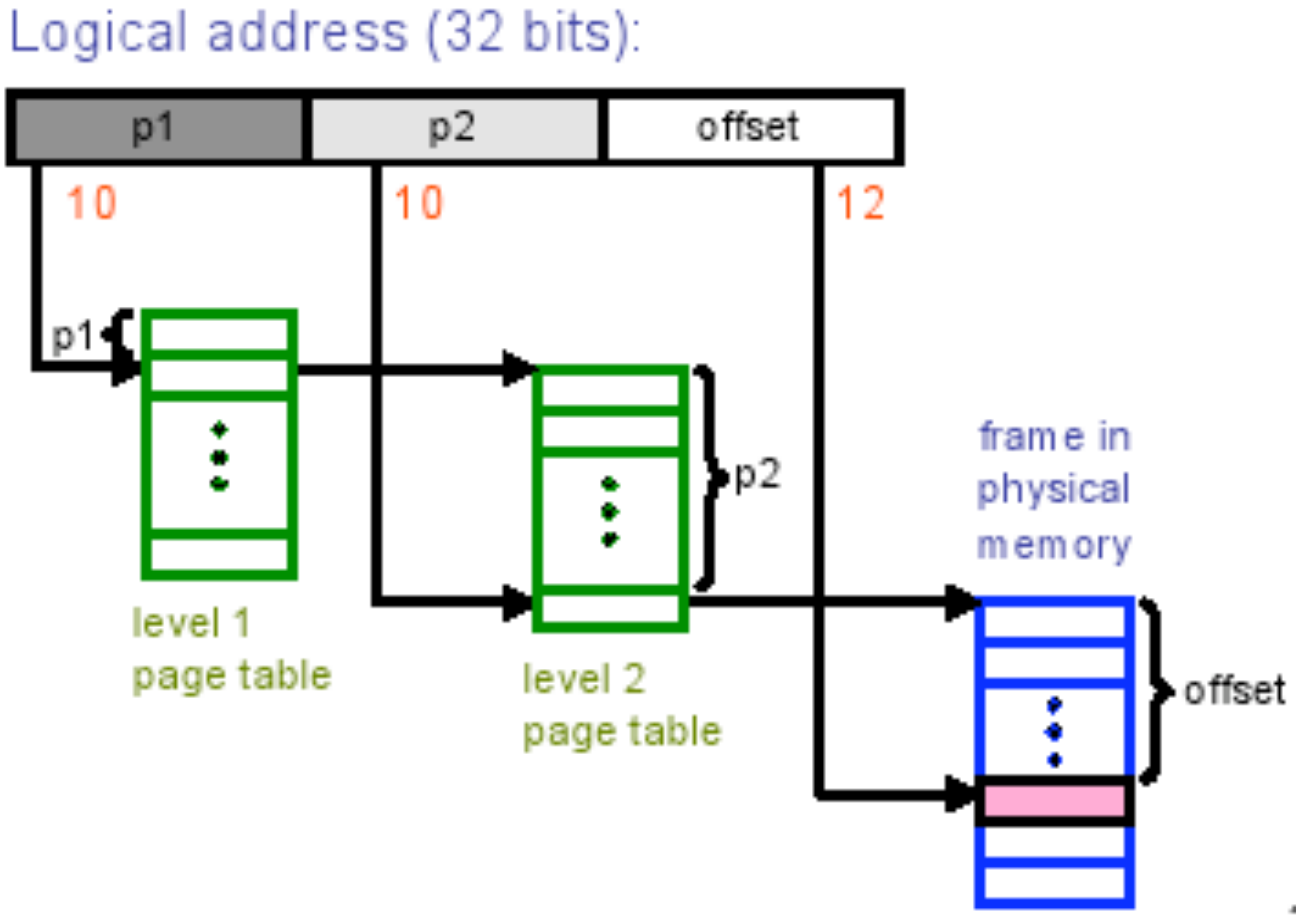
Ilustrasi translasi multilevel page



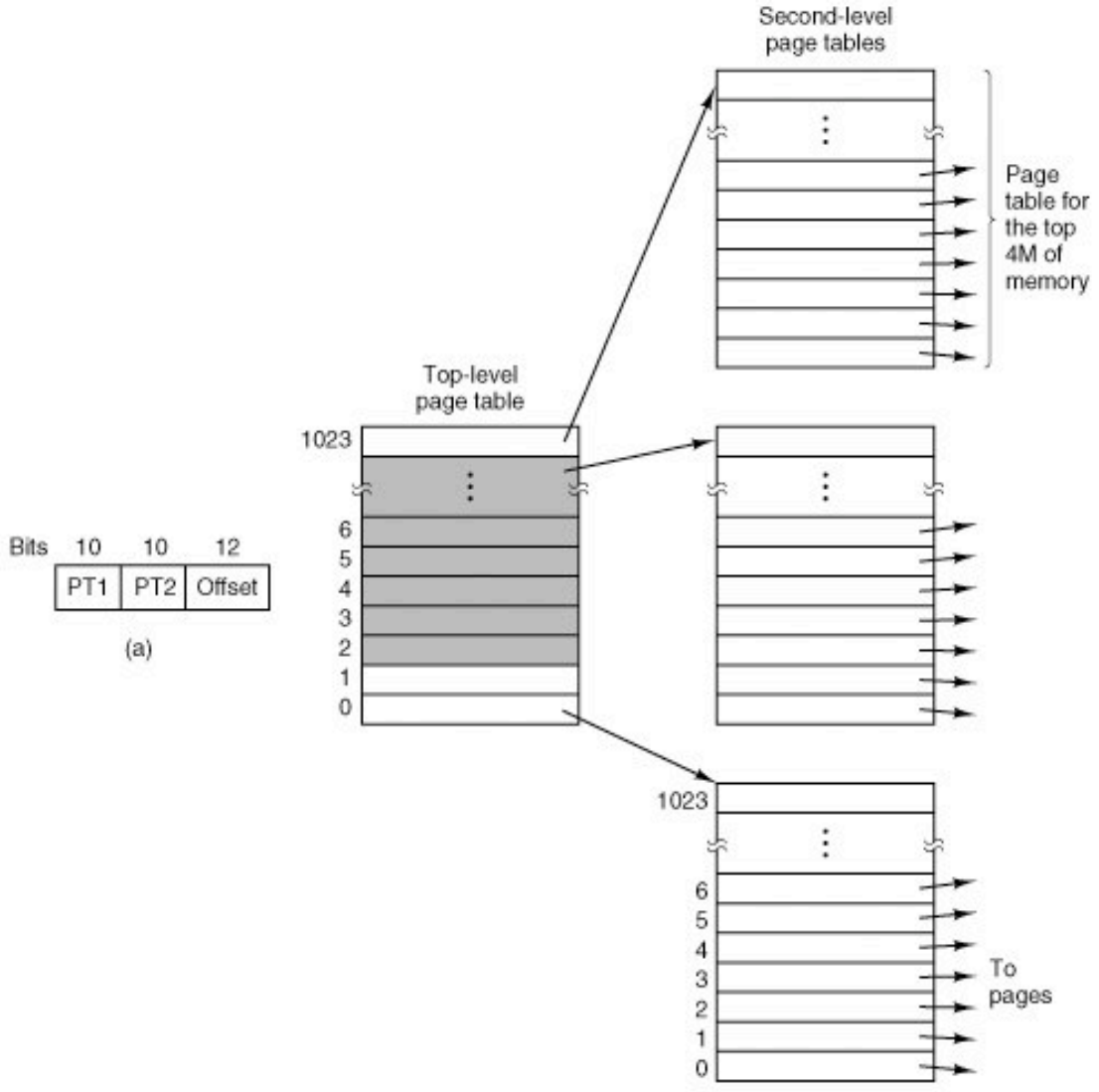




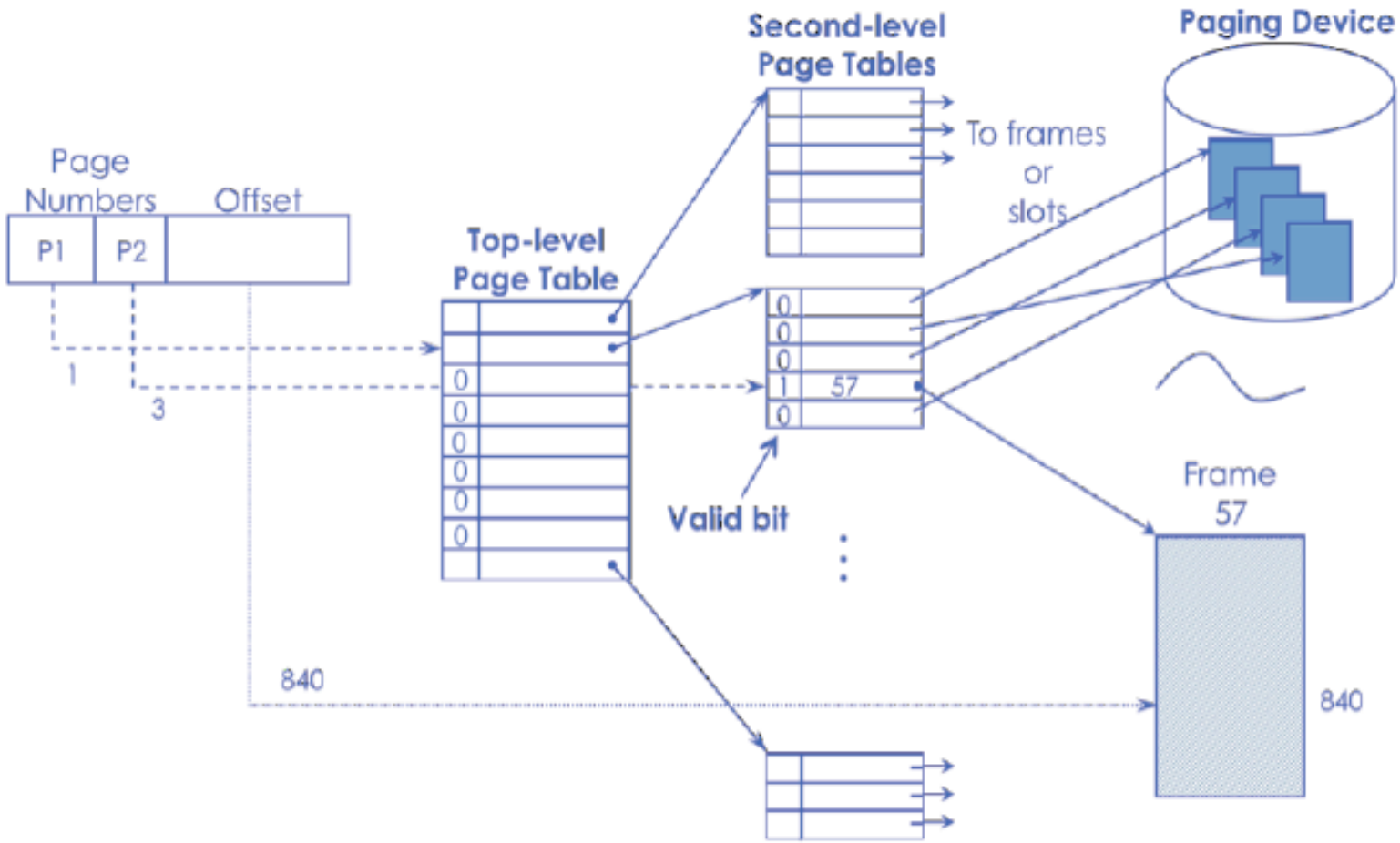
Page Table dua tingkat



Page Table dua tingkat



Contoh dua tingkat page table

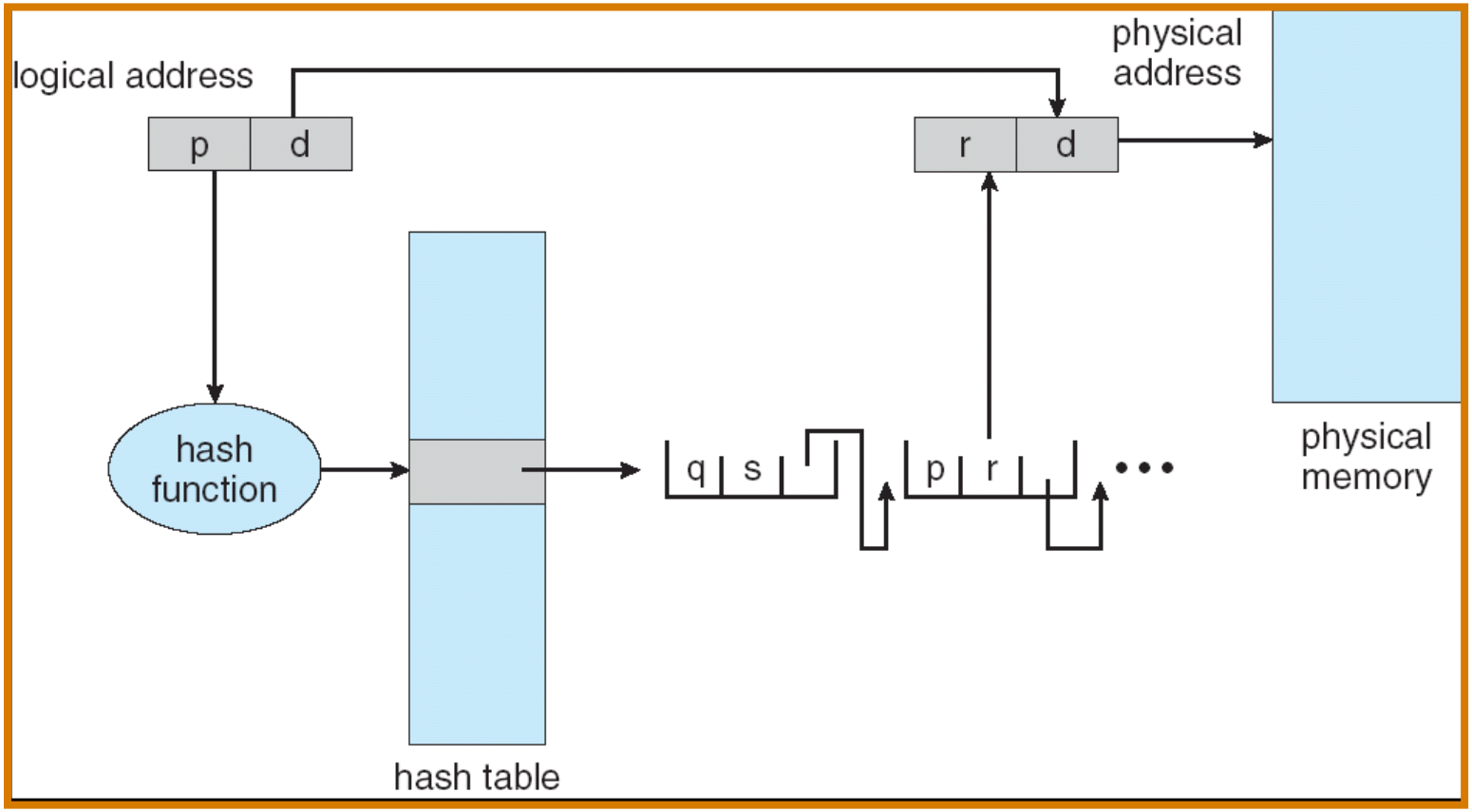


Skema paging dengan dua & tiga tingkat

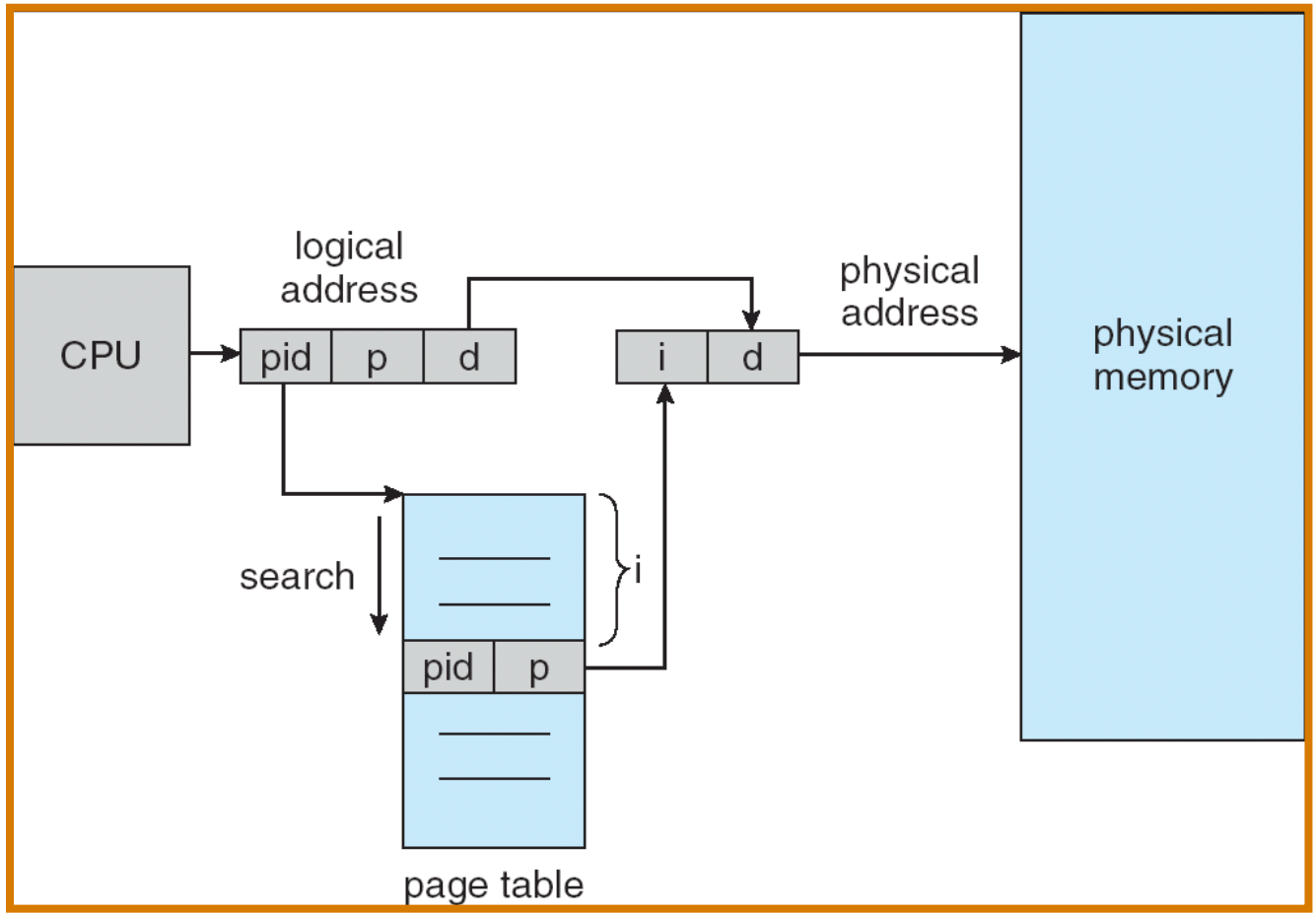
outer page	inner page	offset
p_1	p_2	d
42	10	12

2nd outer page	outer page	inner page	offset
p_1	p_2	p_3	d
32	10	10	12

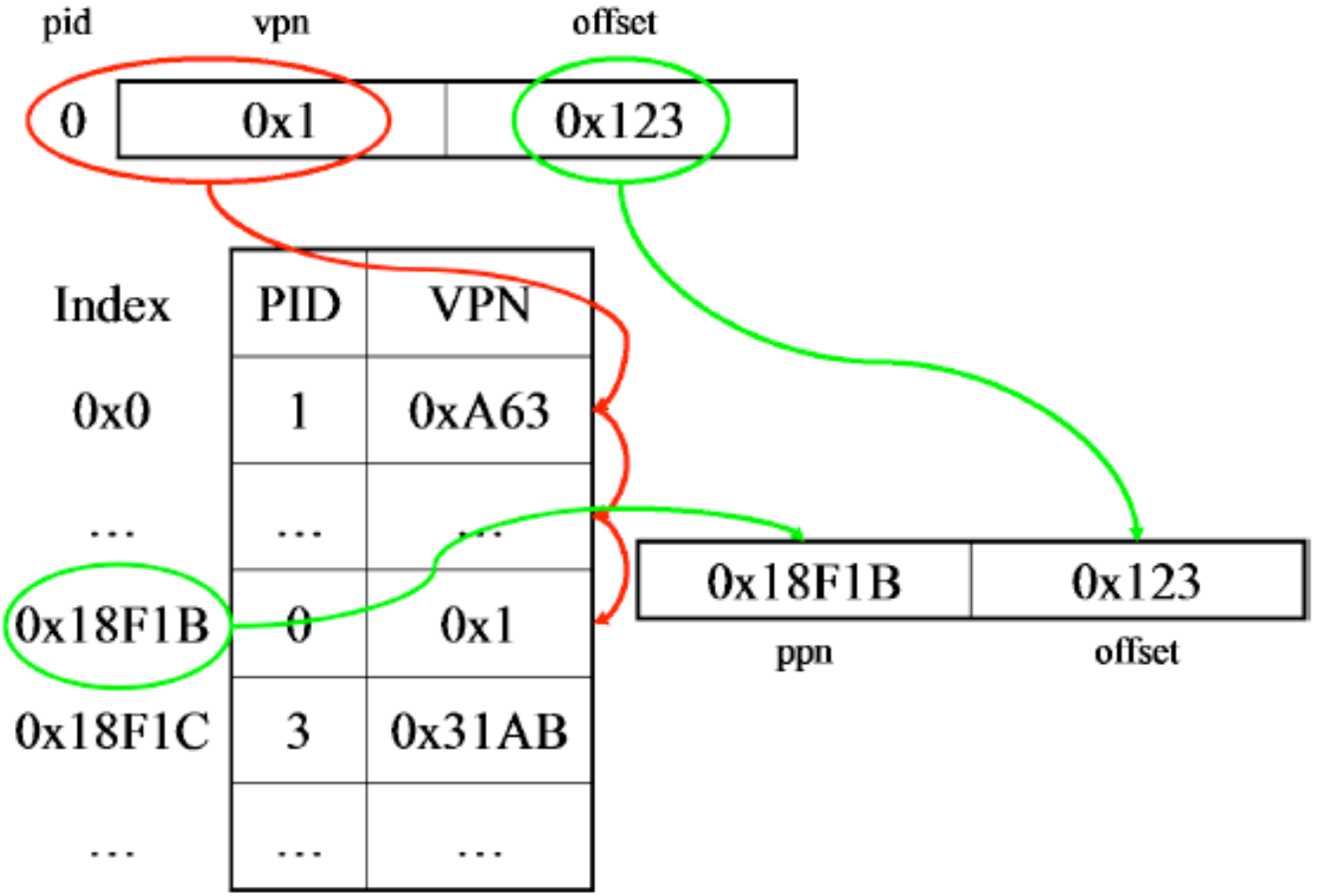
Hashed Page Table



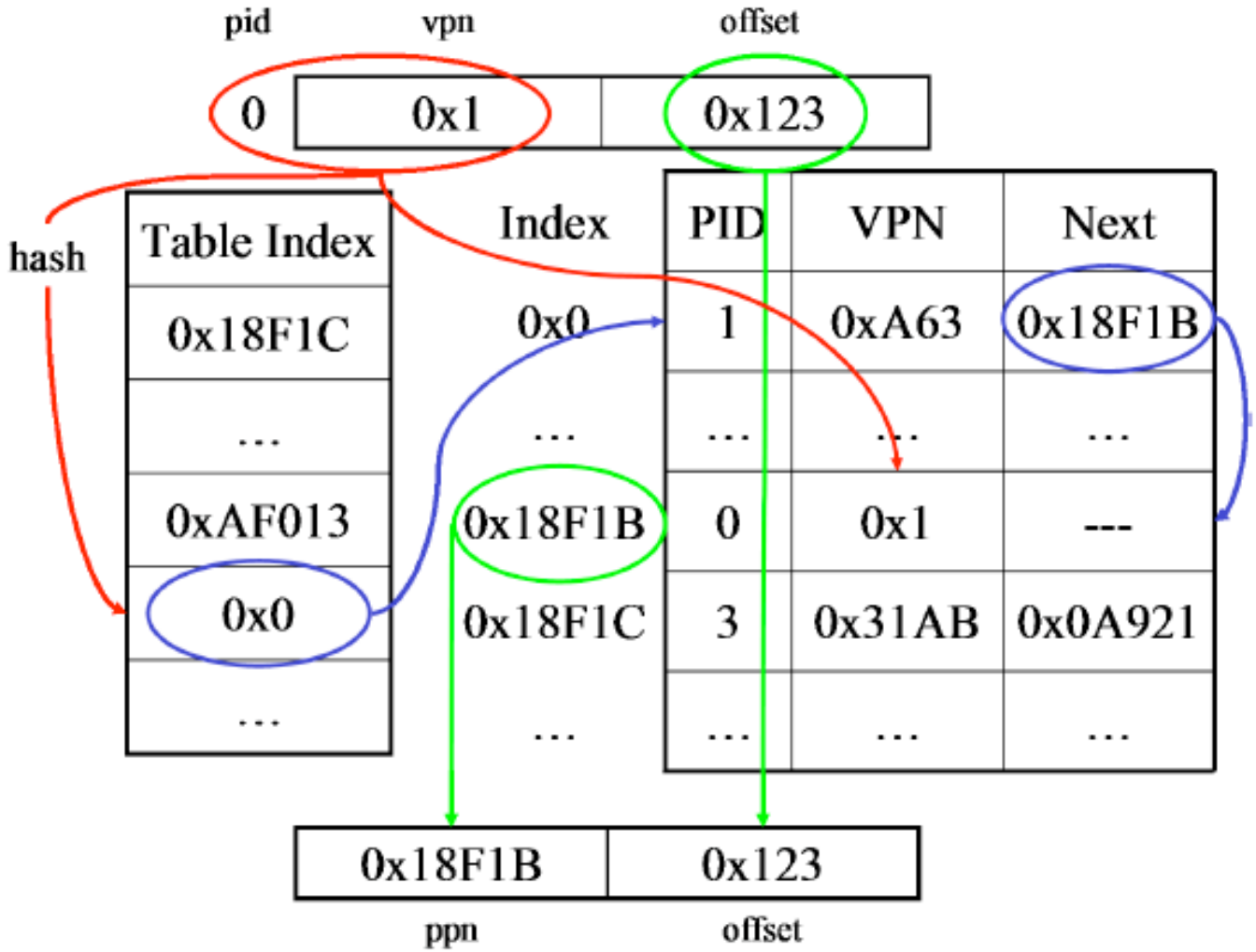
Arsitektur Inverted Page Table



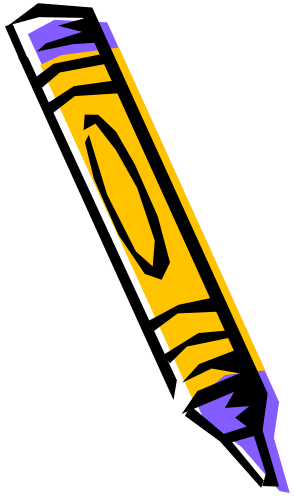
Inverted Page Table



Hashed Inverted Page Table



Pertanyaan



- Apa manfaat dan kerugian menerapkan multilevel page table?
- Bagaimana Hashed-Page Table bekerja?
- Bagaimana Inverted Page Table bekerja?
- Mengapa kadang inverted Page Table dengan Hashed Page Table digabungkan?
- Bagaimana gabungan page table tersebut bekerja?

Latihan

- Jika terdapat sebuah Inverted Page Table (8-entry IPT) pada sistem operasi 32-bit. Ukuran pagenya adalah 2MB. Physical Page Number (PPN) mulai dari 0 hingga 7. Terdapat 3 proses aktif, P1 (PID=1), P2 (PID=2) and P3 (PID=3) yang berjalan pada sistem. IPT menyimpan translasi dari alamat logik ke alamat fisik.

Valid	Process ID (PID)	Virtual Page Number (VPN)
1	1	0x3fe
1	3	0x001
1	2	0x1ad
1	3	0x7fd
1	2	0x3fe
1	1	0x2bf
0	2	0x7fd
1	2	0x0bf

Latihan

a) Berapa besar memori fisik ?

b) Apa alamat fisik (dalam hex) dari alamat logik 0x7fdd8f64 proses P2?

c) Tentukan alamat logik (dalam hex) dan ID Proses dari alamat fisik 0x78e968?

d) Jika diubah dengan menggunakan Page table berapa besar memori yang dibutuhkan? Asumsi : terdapat tambahan 2 bit untuk present dan modified bit

Jawaban

a) Jika terdapat 8 entri pada IPT, maka terdapat 8 frame pada memori fisik
8 entri x 2 Mb= 16 Mb besar memori fisik

b) 2MB → 21 bits page offset
32 – 21 = 11 bits virtual page number (VPN)
7fdd8f64 = 0111 1111 1101 1101 1000 1111 0110 0100
VPN = 0111 1111 110 = 0x3fe → the 5th entry in the IPT → PPN = 100
Physical address = 1001 1101 1000 1111 0110 0100 = 0x9d8f64

c) 0x78e968 = 0111 1000 1110 1001 0110 1000
PPN = 011 = 3 → the 4th entry which has a valid entry with VPN=0x7fd for PID=3
Virtual address = 0x7fd || 1 1000 1110 1001 0110 1000
Virtual address = 1111 1111 1011 1000 1110 1001 0110 1000
Virtual address = 0xffb8e968 of Process P3

d) Karena terdapat 8 page dimemori fisik, maka dibutuhkan 3 bit untuk menampung nomor frame. Satu entri page table terdiri dari: bit Nomor Frame + Present Bit + Modified bit
Total bit = 3 bit + 1 bit +1 bit = 5 bit
Besar memori yang dibutuhkan : jumlah proses x jumlah entri PT x besar satu entri PT
$3 * 2048 * 5bit = 15 \times 2^{11}$ bit = 30Kbits=3.75Kbytes