

Dukungan Sistem Operasi

Pertemuan 12

Oleh :

Riyanto Sigit, ST, M.Kom

Nur Rosyid, ST

Setiawardhana, ST

Hero Yudo Martono, ST

Tujuan

1. Memahami tentang system operasi
2. Memahami tentang penjadwalan proses
3. Memahami Jenis-jenis memori management
Mengerti tentang Swaping dan Partition
4. Memahami konsep Paging, segmentation
5. Mengerti konsep Vitual memori.
6. Mengerti implemantasi pengaturan memori pada powerPC

Translation Lookaside Buffer

- Pada prinsipnya, setiap referensi ke virtual memori terdapat dua jenis akses, yaitu akses untuk mengambil page table entry yang diinginkan dan akses untuk pengambilan datanya.
- Akibat hal ini terjadi penggandaan waktu akses yang menurunkan kinerja.
- Solusi persoalan penggandaan waktu akses adalah dengan menggunakan cache khusus untuk page table entry, yang biasa disebut translation lookaside buffer (TLB).
- Yang harus diperhatikan juga, TLB harus berinteraksi dengan sistem cache memori utama, seperti terlihat pada gambar 5.11.
- Pertama, sistem akan memeriksa TLB untuk mengetahui apakah page table entry yang cocok tersedia, bila ada maka alamat fisik (real) akan dibuat dengan mengkombinasikan nomer frame dengan offset.
- Bila tidak ada, entry diakses dari sebuah page table. Setelah alamat real dibuat, yang berbentuk sebuah tag dan remainder, cache diperiksa untuk mengetahui keberadaan blok yang berisi word tersebut.
- Bila ada, maka akan dikembalikan ke CPU. Sedangkan bila tidak ada, word akan dicari dari memori utama.

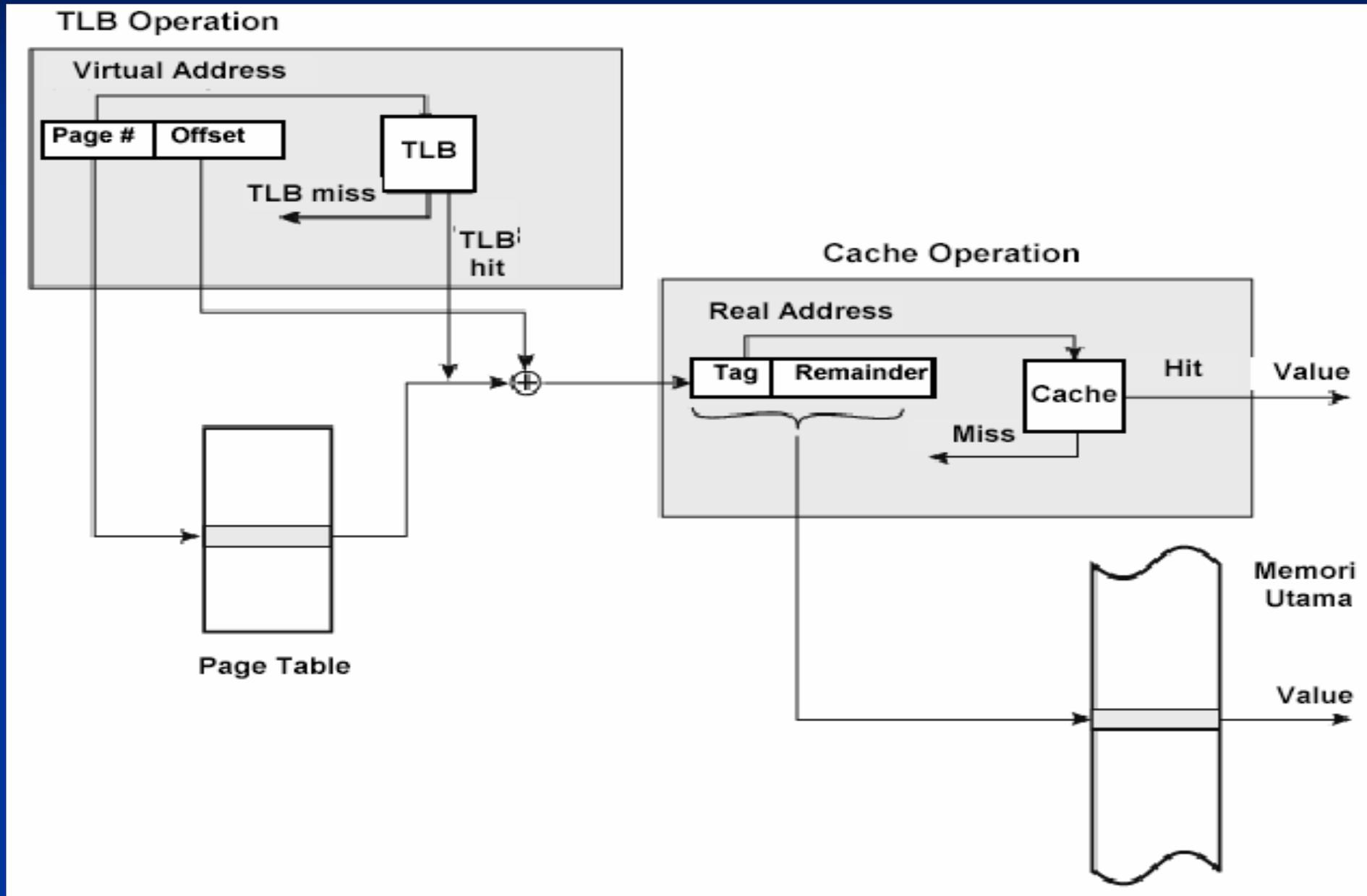
Segmentasi

- Teknik paging tidak dapat diakses atau digunakan pemrogram secara langsung.
- Terdapat teknik manajemen pengalamatan lain yang dapat diakses dan digunakan pemrogram untuk memudahkan dalam mengorganisasikan program dan data, juga sebagai alat untuk privilege dan proteksi yang berkaitan dengan instruksi dan data. Teknik ini dinamakan segmentasi.
- Segmentasi memungkinkan pemrogram untuk menganggap memori sebagai sesuatu yang terdiri dari sejumlah ruang alamat atau segmen. Ukuran segmen variabel dan dinamik

Keuntungan Segmentasi

- Menyederhanakan perkembangan struktur data.
- Memungkinkan program dapat diubah dan dikompilasi ulang secara independen.
- Memungkinkan suatu segmen yang merupakan bagian dari suatu proses digunakan oleh proses lainnya.
- Memungkinkan adanya proteksi, karena pemrogram dapat dengan mudah memberikan akses privilege

Operasi interaksi TLB dan memori cache



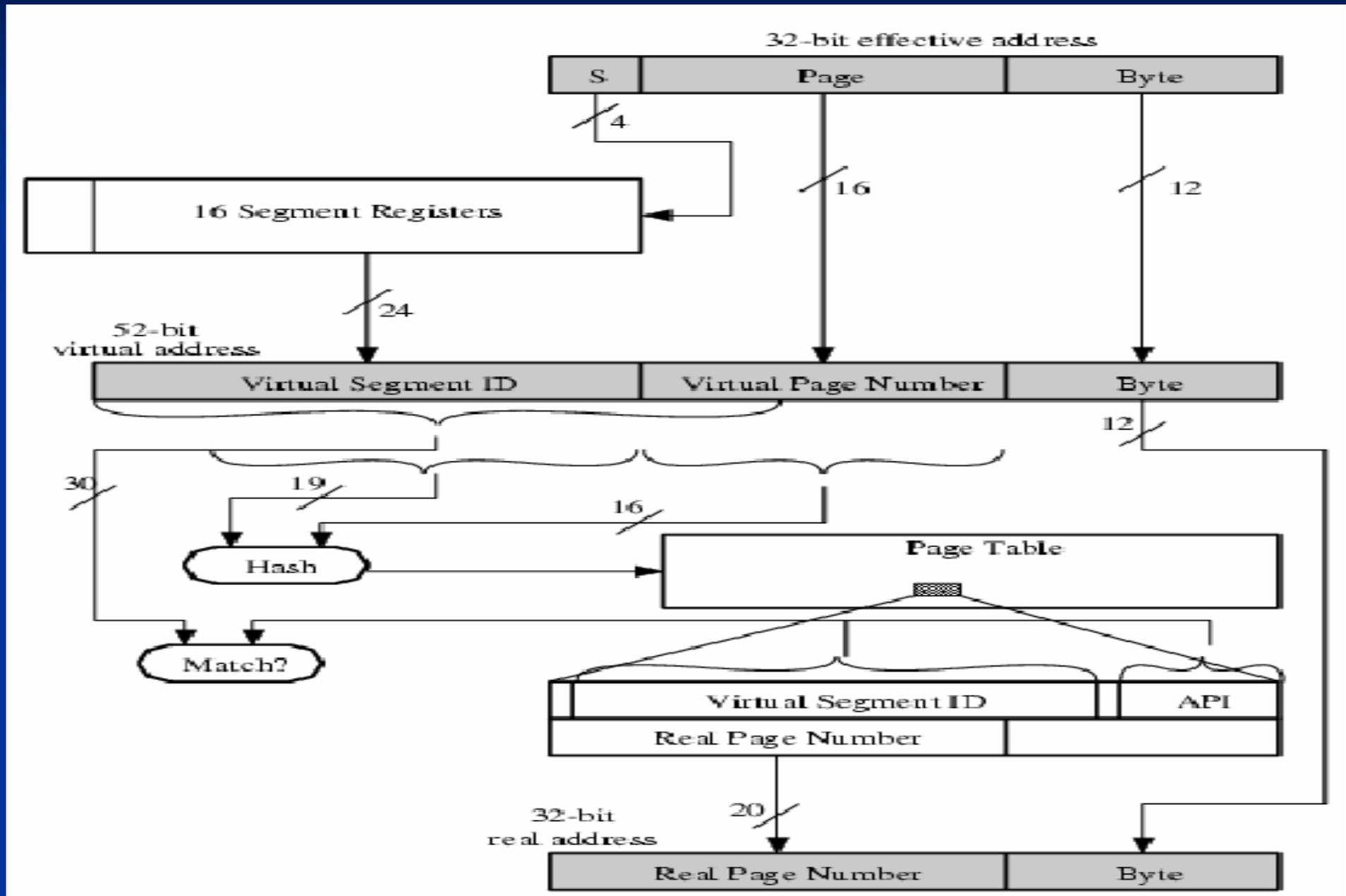
Memori Power PC (1)

- PowerPC menyediakan pengalamatan yang komprehensif.
- Untuk implementasi arsitektur 32 bit menggunakan teknik paging dengan mekanisme segmentasi yang sederhana.
- Bagi implementasi 64 bit digunakan paging dengan mekanisme segmentasi yang lebih baik.
- Selain itu terdapat mekanisme alternatif bagi keduanya, yang dikenal sebagai penerjemahan alamat blok.
- Teknik pengalamatan blok ditujukan untuk menutupi kelemahan teknik paging.
- Dengan pengalamatan blok memungkinkan prosesor untuk memetakan empat buah blok besar memori instruksi dan empat buah blok memori data yang berukuran besar dengan tidak menggunakan mekanisme paging.

Memori Power PC (2)

- PowerPC 32 bit menggunakan alamat efektif 32 bit.
- Alamat tersebut meliputi identifier page 16 bit dan selektor 12 bit.
- Dengan 12 bit dapat mereferensi 4 Kbyte page.
- Empat bit dari alamat digunakan untuk menandakan salah satu dari 16 segmen meliputi bit – bit kontrol akses dan sebuah identifier 24 bit, sehingga alamat efektif 32 bit memetakan ke dalam alamat virtual 52 bit.

Mekanisme pengalamatan PowerPC 32 bit



PowerPC

- PowerPC menggunakan inverted page table tunggal.
- Alamat virtual digunakan untuk mengindekskan ke dalam page table.
- Nomer virtual page diisikan ke bagian kiri (sisi yang paling berarti) dengan tiga buah nol untuk membentuk bilangan 19 bit.
- Kemudian bit per bit exclusive-OR dihitung ke bilangan tersebut dan 19 bit paling kanan dari ID segmen virtual untuk membentuk kode hash 19 bit.
- Tabel diatur menjadi n kelompok yang terdiri dari 8 buah entry. Dari 10 sampai 19 bit kode hash (tergantung pada ukuran page table) digunakan untuk memilih salah satu kelompok di dalam tabel.
- Kemudian hardware manajemen memori malacak 8 entry dalam kelompok untuk dicocokkan dengan alamat virtualnya

PowerPC

- Dalam mencari kecocokan tersebut, setiap page table entry meliputi ID segmen virtual
- 6 bit paling kiri dari nomer virtual page, yang disebut abbreviated page index (karena sedikitnya 10 bit dari nomer virtual page 16 bit selalu berpartisipasi di dalam hash untuk memilih kelompok page table entry, hanya bentuk singkat nomer virtual page yang perlu dibawa ke page table entry agar cocok secara unik dengan alamat virtual).
- Bila terdapat kecocokan, maka nomer real page 20 bit dari alamat dirangkaikan dengan 12 bit terendah alamat efektif untuk membentuk alamat fisik 32 bit untuk diakses.

PowerPC

- Bila tidak terdapat kecocokan maka kode hash dikomplemenkan untuk menghasilkan indeks page table yang baru yang berada pada posisi relatif sama di sisi lain tabel.
- Kemudian kelompok ini dilacak untuk dicocokkan. Bila tidak ada yang cocok lagi, sebuah page fault interrupt akan terjadi
- Pada manajemen pengalamatan 64 bit PowerPC dirancang untuk kompatibel dengan 32 bit. Prinsipnya sama dengan mekanisme pengalamatan 32 bit.

Kesimpulan

- Operating sistem bertugas sebagai penghubung antara user dan perangkat keras, agar user dapat menggunakan perangkat keras secara Nyaman dan efisien
- Operating sistem mengatur sumber daya perangkat keras.
- Penjadwalan kerja perangkat keras oleh operating sistem dapat mengoptimalkan kecepatan CPU.
- Ada beberapa istilah dalam manajemen memori, yaitu: swapping, partitioning, paging, virtual memory.
- Memori virtual merupakan bagian dari secondary yang dianggap sebagai memori utama

Latihan Soal

- Jelaskan fungsi Sistem Operasi!
- Dalam membedakan jenis sistem operasi adalah berdasarkan sifat eksekusinya, sebutkan dan jelaskan jenisnya!
- Dalam sistem batch terdapat Single-programming & Multi-programming Jelaskan 2 hal tersebut!
- Jelaskan tentang translation lookaside buffer!
- Jelaskan tentang swapping, partitioning, paging, virtual memory!