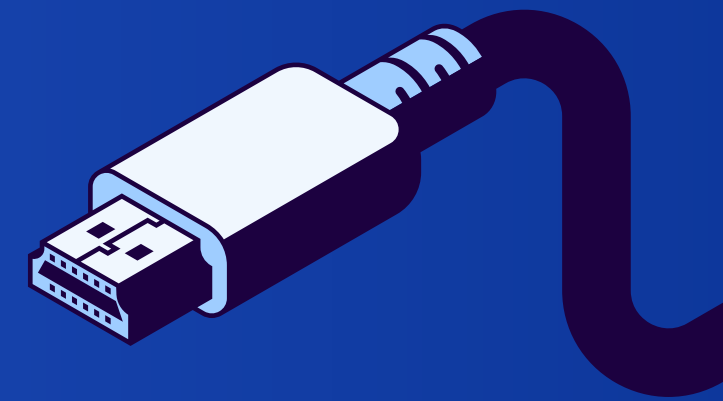
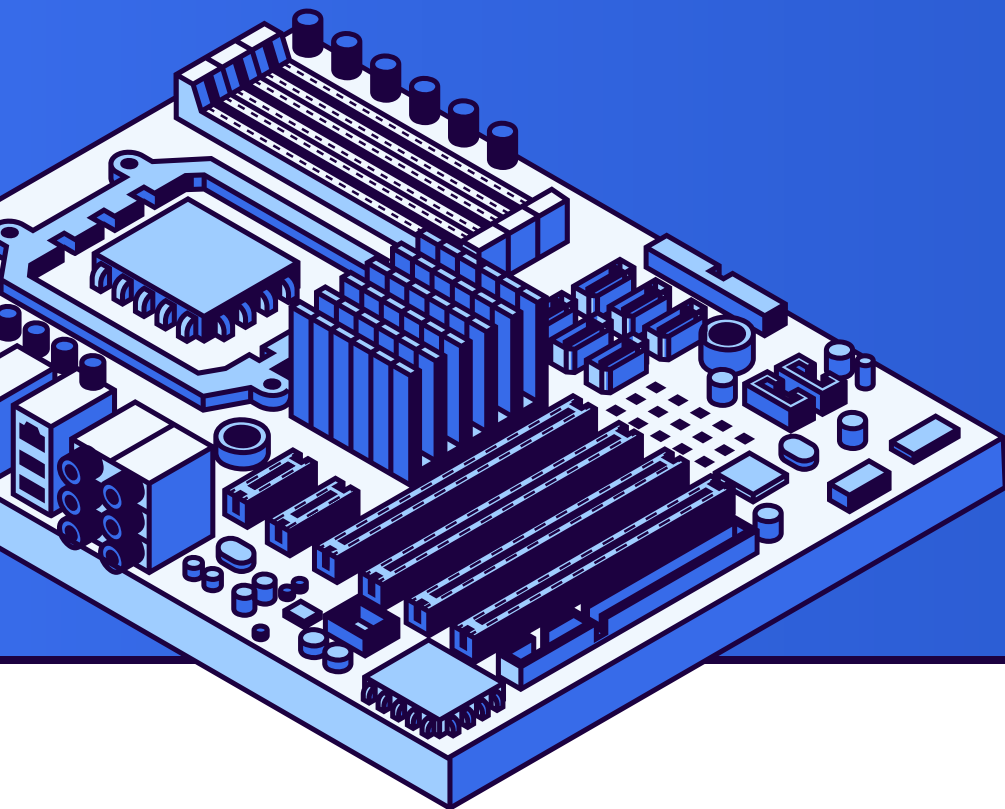


Virtual Memory



Kelompok 7 :

- Gusti Panji Widodo (2407113145)
- Nahda Zahirah (2407113190)
- Selviana Ananda Permata Tanjung (2407112863)

Virtual Memory

DEFINISI

Memori Virtual adalah sebuah teknik manajemen memori yang memisahkan ruang alamat logis (pandangan memori dari sisi program) dari ruang alamat fisik (RAM yang sebenarnya terpasang). Teknik ini memberikan ilusi kepada setiap proses seolah-olah ia memiliki memori yang sangat besar dan privat, padahal kapasitasnya bisa jauh melampaui RAM fisik.

Mekanisme ini bekerja dengan menggunakan penyimpanan sekunder (seperti hard disk) sebagai backing store dan memindahkan data antara RAM dan disk sesuai kebutuhan, sebuah proses yang dikenal sebagai paging atau swapping.



Tujuan Utama Virtual Memory

MENJALANKAN PROGRAM BESAR

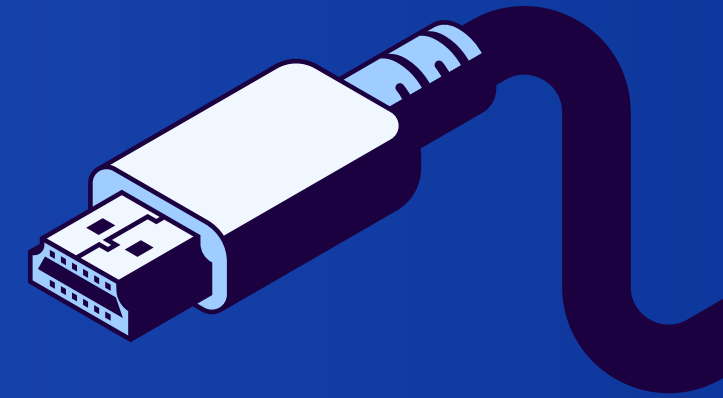
Memungkinkan eksekusi program yang ukurannya lebih besar dari RAM fisik dengan hanya menyimpan bagian yang aktif digunakan di memori.

MENINGKATKAN MULTIPROGRAMMING

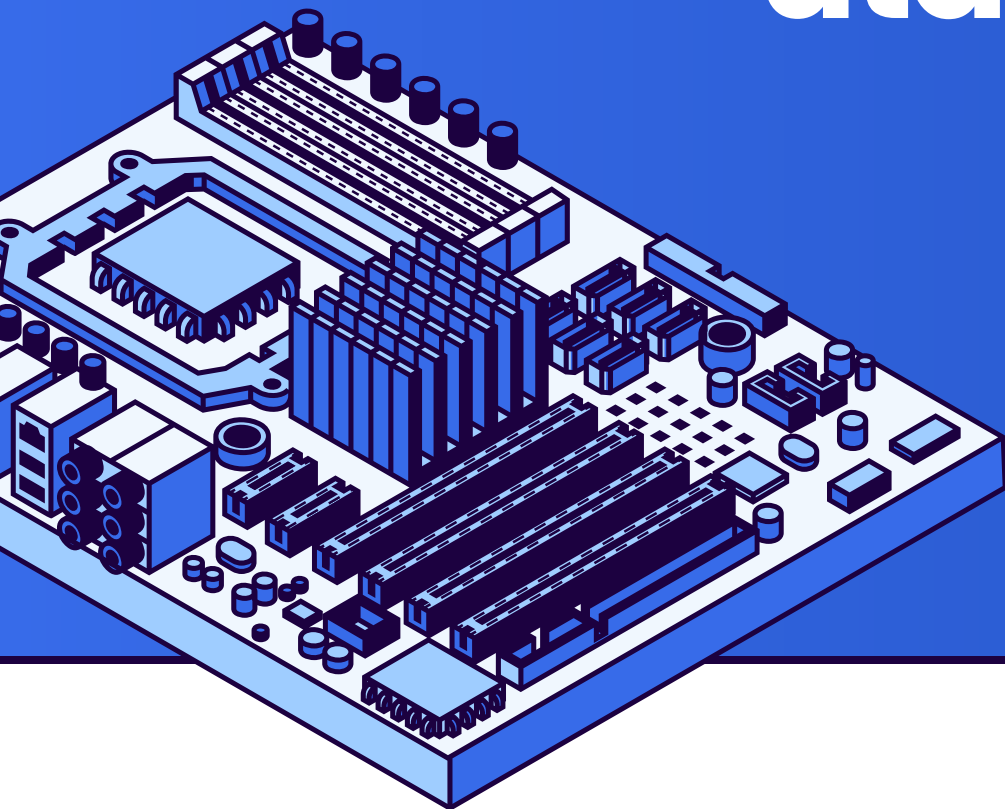
Lebih banyak proses dapat dimuat ke memori secara bersamaan, sehingga meningkatkan utilisasi CPU.

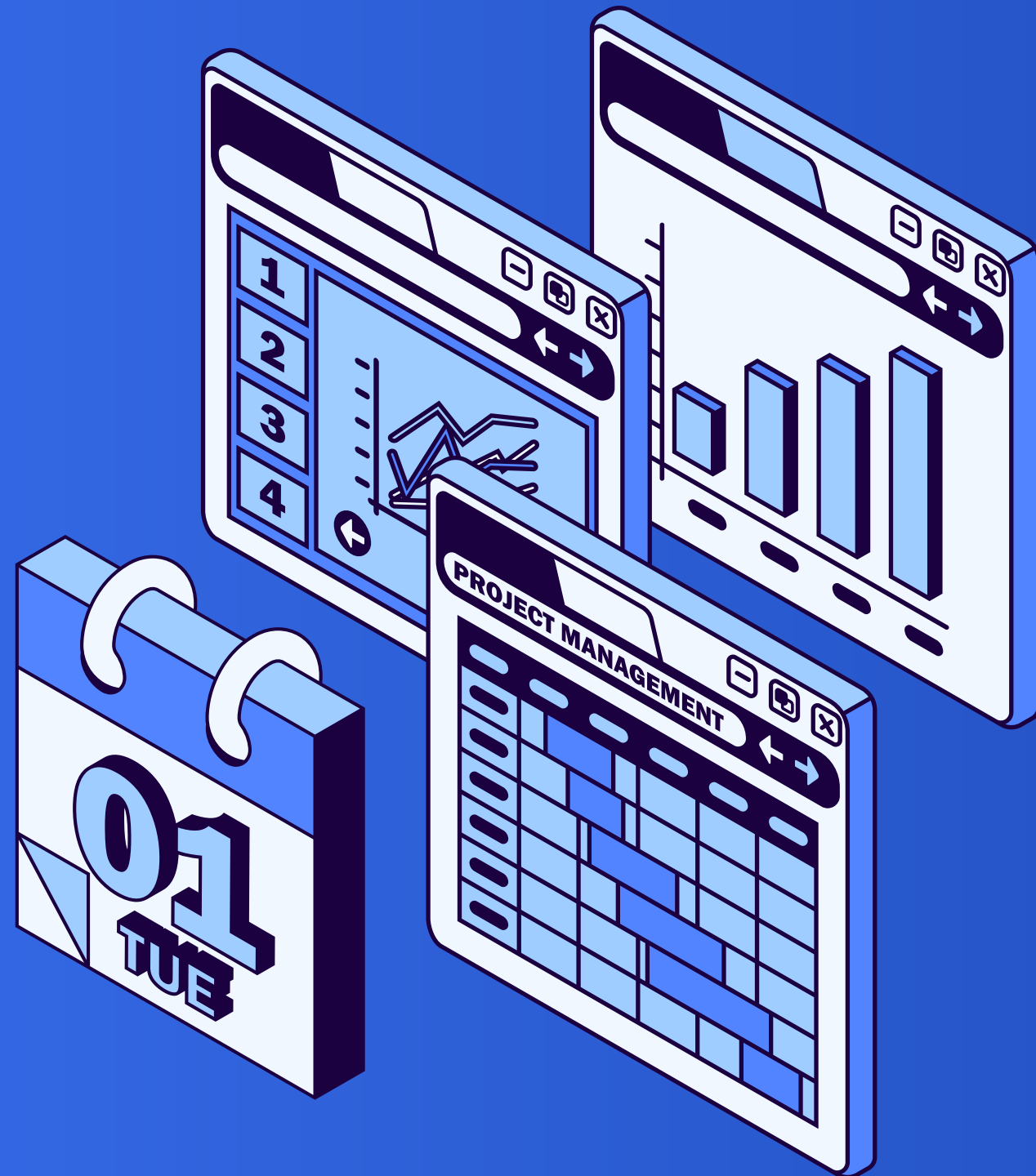
PROTEKSI MEMORI

Mengisolasi ruang alamat antar proses, mencegah satu proses mengganggu proses lainnya, yang sangat penting untuk stabilitas dan keamanan sistem.



**Untuk memetakan alamat logis ke fisik,
sistem operasi menggunakan dua metode
utama: Paging dan Segmentasi.**





Paging

Paging adalah teknik yang membagi memori logis menjadi blok-blok berukuran tetap yang disebut halaman (pages), dan memori fisik menjadi blok-blok berukuran sama yang disebut bingkai (frames). Sistem operasi menggunakan Tabel Halaman (Page Table) untuk setiap proses, yang berfungsi memetakan setiap halaman virtual ke bingkai fisik. Proses translasi alamat ini dipercepat oleh perangkat keras khusus yang dikenal sebagai Memory Management Unit (MMU).

Kelebihan Metode Paging

Mengeliminasi fragmentasi eksternal karena semua unit alokasi (bingkai) berukuran sama.


Manajemen alokasi menjadi sederhana.

Transparan bagi pemrogram.




Kekurangan Metode Paging

Rentan terhadap fragmentasi internal, yaitu ruang yang terbuang di dalam halaman terakhir sebuah proses.

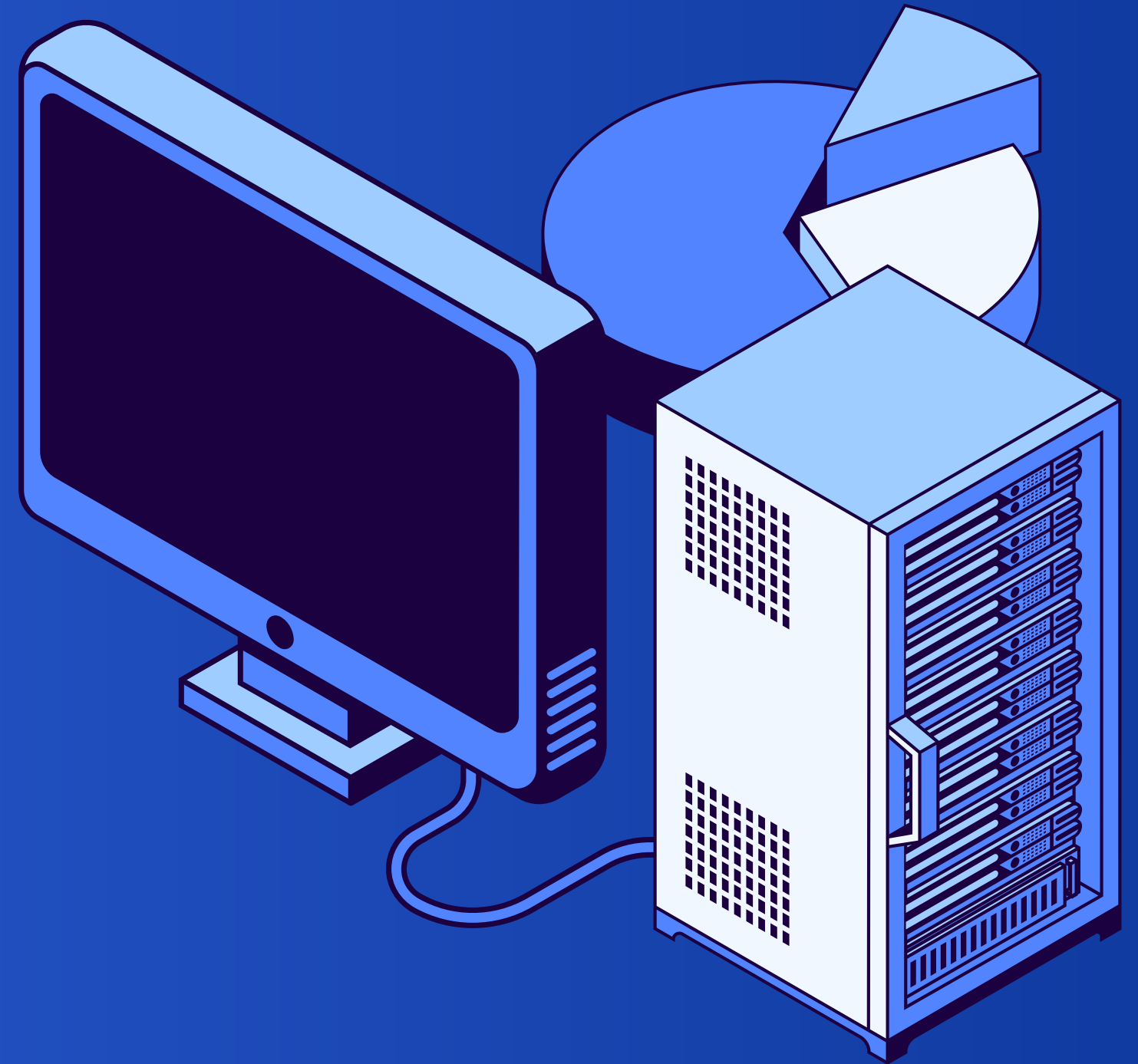


Memerlukan ruang memori untuk menyimpan tabel halaman, yang bisa menjadi besar.



Segmentasi

Segmentasi membagi memori logis menjadi unit-unit berukuran variabel yang disebut segmen, yang sesuai dengan struktur logis program (misalnya, segmen kode, data, dan stack). Mekanismenya adalah alamat logis terdiri dari <nomor segmen, offset>, dan Tabel Segmen (Segment Table) menyimpan alamat dasar (lokasi awal) serta batas (panjang) dari setiap segmen untuk translasi alamat.




Kelebihan Metode Segmentasi

Mendukung struktur logis program, memudahkan proteksi dan berbagi memori antar proses.


Tidak ada fragmentasi internal karena segmen dialokasikan sesuai kebutuhan.

Kekurangan Metode Segmentasi

Menyebabkan fragmentasi eksternal, di mana ruang kosong di memori menjadi terpecah-pecah dan sulit digunakan.



Alokasi memori lebih kompleks karena ukuran segmen yang bervariasi.



Page Replacement Algorithm

Ketika terjadi page fault (halaman yang dibutuhkan tidak ada di RAM) dan tidak ada bingkai kosong, sistem operasi harus memilih halaman "korban" untuk diganti. Algoritma penggantian halaman berfungsi untuk menentukan halaman mana yang akan dikeluarkan, dengan tujuan meminimalkan jumlah page fault.

Ada beberapa algoritma penggantian halaman utama:



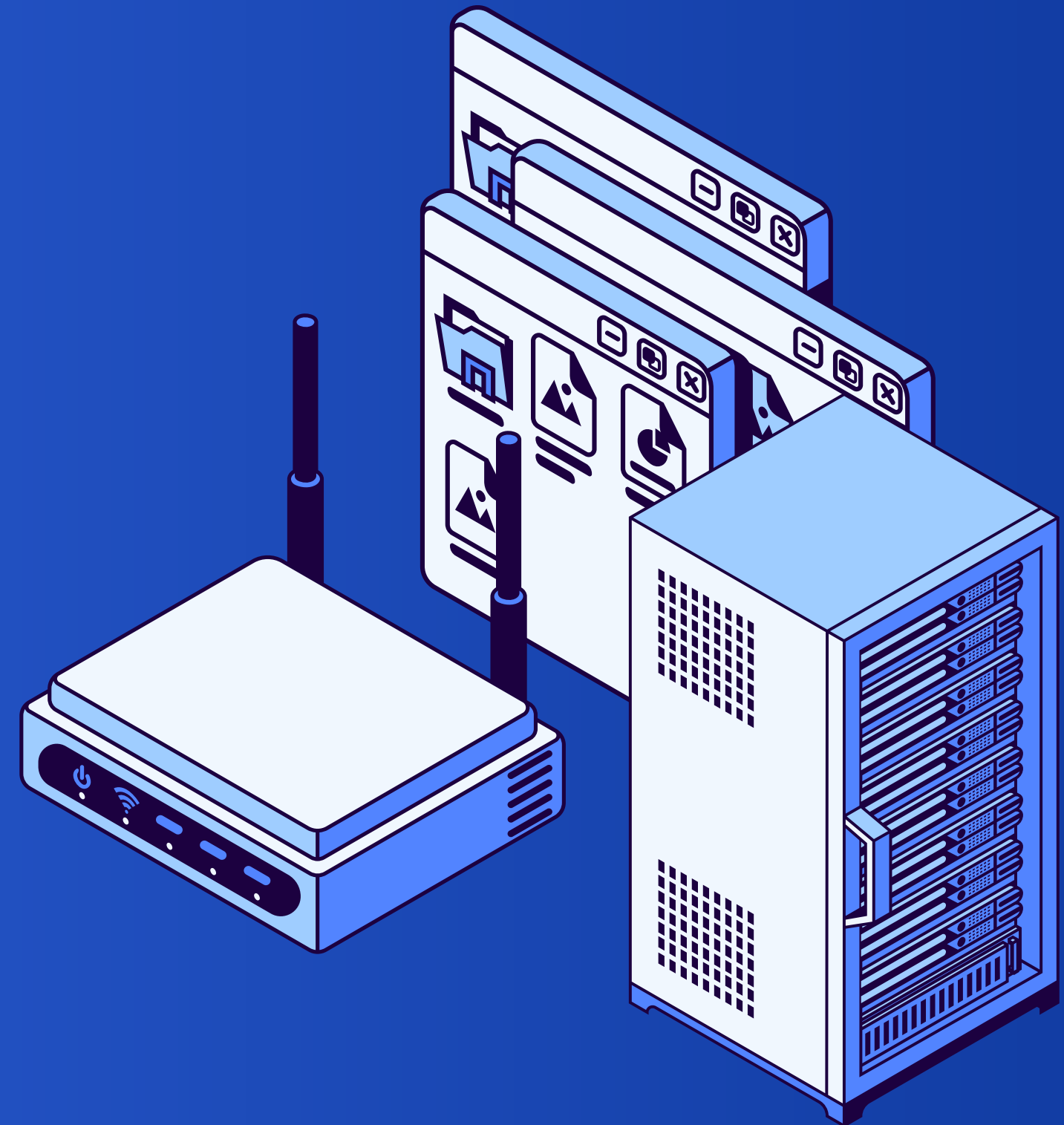


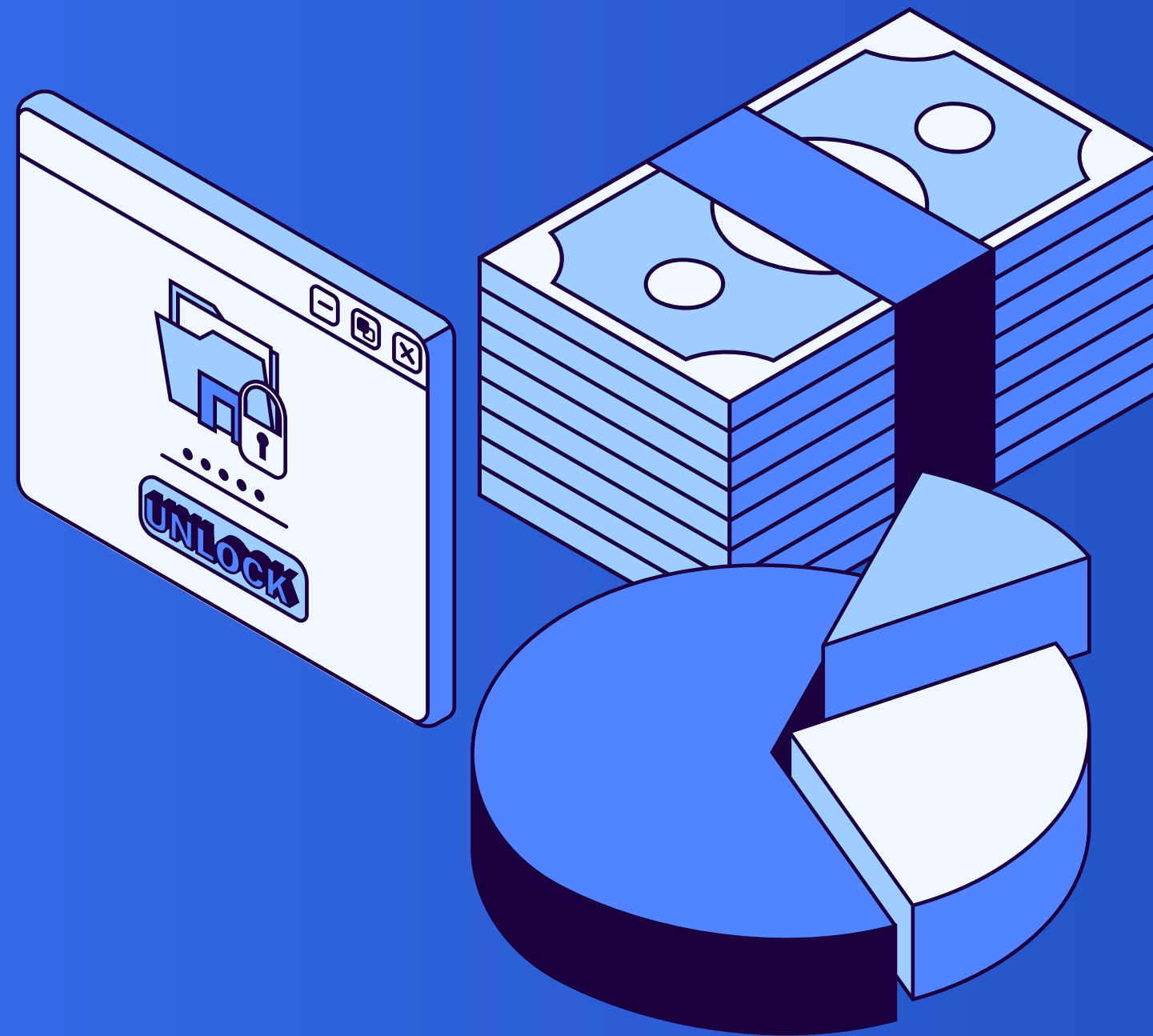
First-In, First Out (FIFO)

Prinsipnya adalah mengganti halaman yang paling lama berada di memori. Algoritma ini paling sederhana untuk diimplementasikan, namun kinerjanya seringkali buruk dan dapat mengalami Anomali Belady, di mana penambahan bingkai justru meningkatkan page fault.

Optimal (OPT/MIN)

Algoritma ini mengganti halaman yang tidak akan digunakan untuk periode waktu terlalu lama di masa depan. Meskipun menghasilkan jumlah page fault terendah dan menjadi tolok ukur teoretis, algoritma ini mustahil diimplementasikan karena memerlukan pengetahuan masa depan.





Least Recently Used

Prinsipnya adalah mengganti halaman yang paling lama tidak digunakan, berdasarkan asumsi bahwa halaman yang baru saja digunakan kemungkinan akan digunakan lagi (prinsip lokalitas) .

Kinerja pada proses ini sangat baik dan mendekati Optimal, tetapi implementasinya lebih kompleks dan memerlukan dukungan perangkat keras atau overhead perangkat lunak yang signifikan .

Thrashing

Thrashing adalah kondisi di mana sistem menghabiskan lebih banyak waktu untuk memindahkan halaman (paging) daripada menjalankan instruksi program. Hal ini terjadi ketika sebuah proses tidak memiliki cukup bingkai memori untuk menampung halaman-halaman yang aktif digunakannya (dikenal sebagai working set).



Penyebab Thrashing

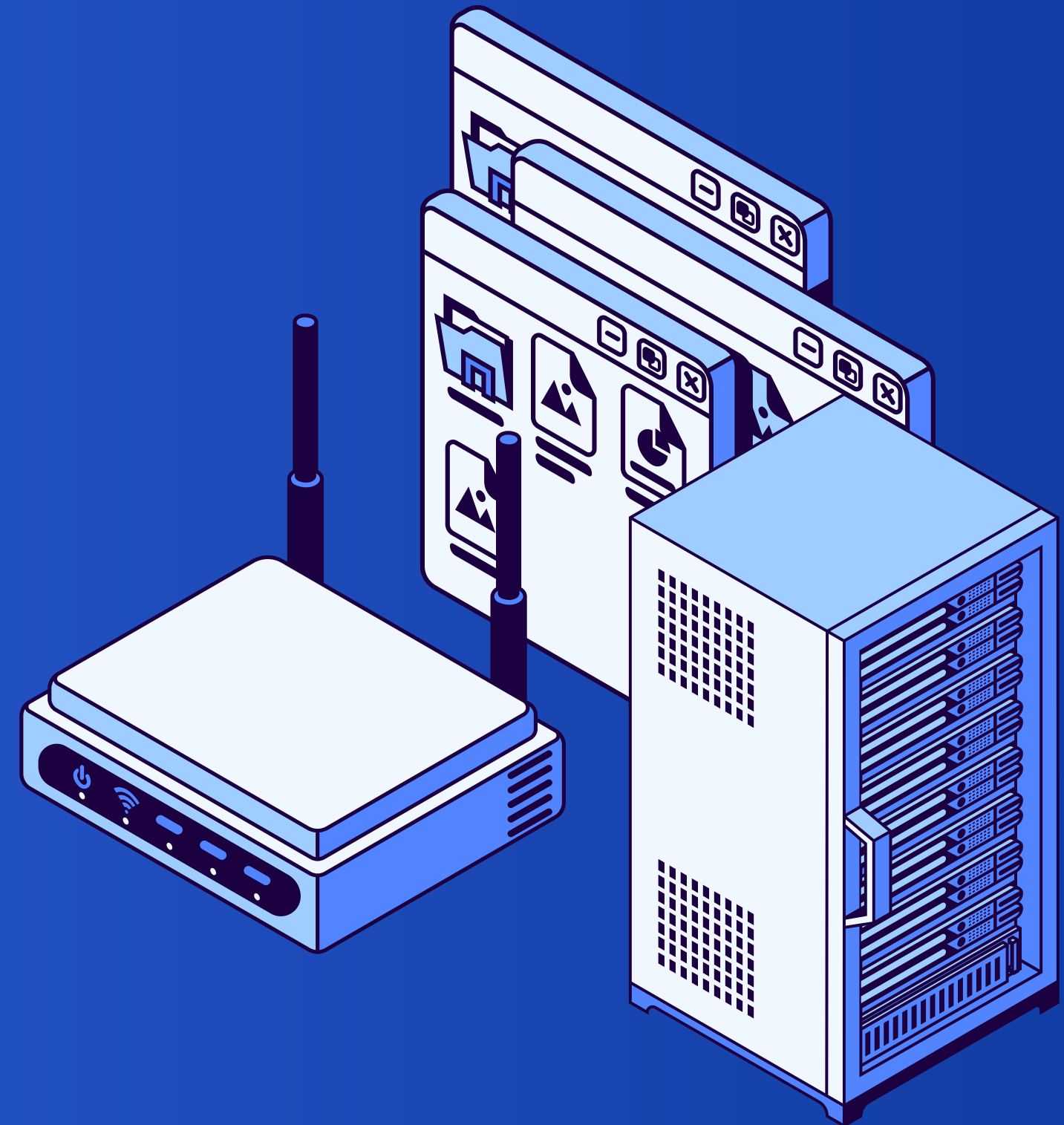
Tingkat page fault yang sangat tinggi menyebabkan CPU lebih banyak menganggur menunggu I/O disk. Sistem operasi yang salah mengartikan utilisasi CPU yang rendah sebagai kurangnya beban kerja akan menambah lebih banyak proses, yang justru memperparah perebutan memori dan menyebabkan thrashing yang lebih parah.



Pencegahan

○ Model Set Kerja (Working-Set Model)

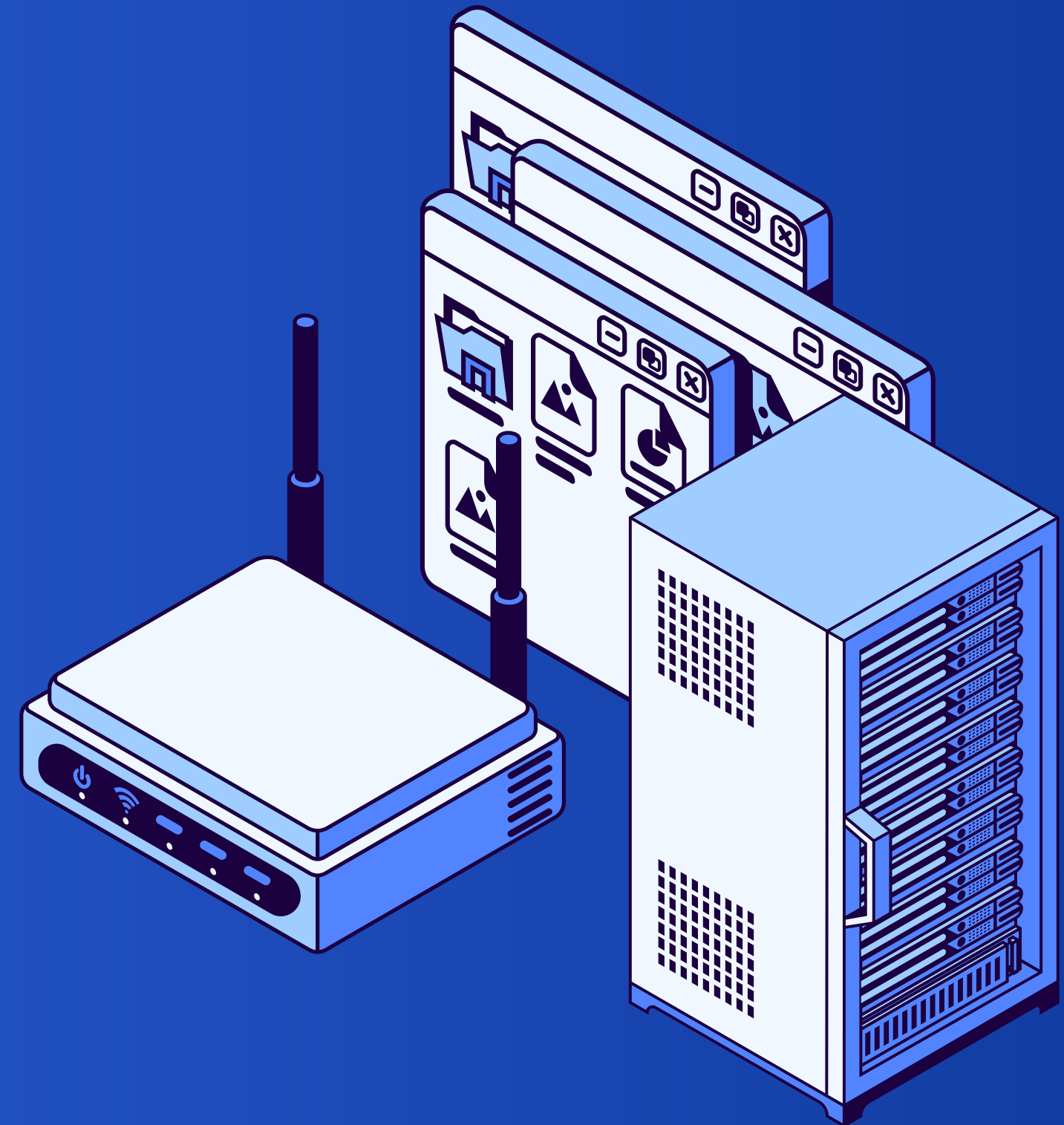
Sistem operasi memonitor halaman-halaman yang aktif digunakan oleh setiap proses (set kerjanya) dan memastikan set tersebut berada di memori sebelum proses diizinkan berjalan. Jika total kebutuhan memori melebihi RAM yang tersedia, beberapa proses akan ditangguhkan.



Pencegahan

○ Frekuensi Page-Fault (PFF)

Mengontrol alokasi bingkai secara dinamis. Jika tingkat page fault suatu proses terlalu tinggi, ia diberi lebih banyak bingkai. Jika terlalu rendah, bingkainya dapat diambil kembali.



Kesimpulan

Memori virtual adalah teknik manajemen memori esensial dalam sistem operasi modern yang memisahkan ruang alamat logis dan fisik, memungkinkan eksekusi program besar, meningkatkan multiprogramming, dan menyediakan proteksi memori melalui mekanisme paging atau swapping. Meskipun paging membagi memori menjadi blok tetap dan menghindari fragmentasi eksternal tetapi rentan terhadap fragmentasi internal, sementara segmentasi menggunakan unit variabel yang sesuai dengan struktur program dan menghindari fragmentasi internal tetapi dapat menyebabkan fragmentasi eksternal. Untuk mengelola page fault, sistem operasi menggunakan algoritma penggantian halaman seperti FIFO, Optimal, dan LRU, dengan LRU menawarkan kinerja terbaik. Penting untuk mencegah thrashing, kondisi di mana sistem terlalu banyak menghabiskan waktu untuk paging, melalui model set kerja dan frekuensi page-fault, guna menjaga stabilitas dan efisiensi sistem multitasking yang kompleks.

Terimakasih!

KELOMPOK 7 SISTEM INFORMASI