

UAS Sistem Operasi

Aldi Maulana Iqbal – 2021080122

Apa yang kamu ketahui tentang Deadlock System? Jelaskan!

Deadlock dapat diartikan sebagai kebuntuan proses, dimana proses tidak dapat melanjutkan eksekusi karena sumber daya yang dibutuhkan tidak tersedia. Deadlock terjadi ketika dua atau lebih proses saling menunggu sumber daya yang digunakan oleh proses lain, sehingga tidak ada proses yang dapat berjalan. Hal ini dapat menyebabkan kerusakan sistem dan harus diatasi dengan cara yang tepat untuk menjaga kestabilan sistem.

Ada beberapa faktor yang dapat menyebabkan terjadinya deadlock dalam sistem operasi:

1. Kondisi Mutual Exclusion: salah satu sumber daya harus dapat digunakan hanya oleh satu proses pada suatu waktu.
2. Kondisi Hold and Wait: sebuah proses harus memegang salah satu sumber daya sementara menunggu sumber daya lain.
3. Kondisi No Preemption: sumber daya yang digunakan oleh proses tidak dapat diambil kembali oleh sistem operasi sebelum proses tersebut selesai menggunakannya.
4. Kondisi Circular Wait: dua atau lebih proses saling menunggu sumber daya yang digunakan oleh proses lain dalam urutan yang saling terkait.
5. Penggunaan sumber daya yang tidak efisien: Proses yang mengambil sumber daya yang tidak diperlukan atau meninggalkan sumber daya yang digunakan dapat menyebabkan deadlock.
6. kesalahan programmer dalam menangani sumber daya, seperti tidak melepaskan sumber daya yang tidak digunakan atau tidak mengecek apakah sumber daya yang dibutuhkan tersedia sebelum mengambilnya.
7. Konflik akses sumber daya dalam sistem yang didistribusikan, seperti dalam sistem jaringan yang saling terkait.
8. pemakaian sumber daya yang tidak dapat dibagi dan dibutuhkan oleh beberapa proses sekaligus.

Sebutkan dan jelaskan metode beserta pencegahan deadlock system!

Metode yang digunakan untuk mencegah atau mengatasi deadlock pada sistem adalah:

1. Metode prevensi (prevention) yaitu mencegah terjadinya deadlock sebelum terjadi dengan cara mengatur alokasi sumber daya.
2. Metode deteksi (detection) yaitu mendeteksi terjadinya deadlock setelah terjadi dan mengambil tindakan untuk mengatasinya.
3. Metode evakuasi (evacuation) yaitu mengeluarkan proses yang terlibat dalam deadlock dari sistem.

Pencegahan deadlock dapat dilakukan dengan beberapa cara, diantaranya adalah:

1. Alokasi sumber daya secara statis sebelum proses dimulai.
2. Alokasi sumber daya secara dinamis saat proses berjalan.
3. Penggunaan algoritma pengalokasian sumber daya yang konsisten, seperti algoritma Banker pada sistem operasi.

4. Penggunaan sistem antrian prioritas untuk menentukan urutan proses yang akan mendapatkan sumber daya.
5. Pembatasan jumlah sumber daya yang dapat digunakan oleh suatu proses.

Sebutkan dan jelaskan mengenai arsitektur pengolahan file!

Arsitektur pengolahan file dapat dibagi menjadi dua jenis utama: arsitektur berbasis file dan arsitektur berbasis database.

Arsitektur berbasis file

Dalam arsitektur ini, file-file yang dibutuhkan disimpan dalam sistem file yang terpisah dari aplikasi yang menggunakannya. Aplikasi mengakses file-file tersebut melalui sistem file yang tersedia di komputer. Keuntungan dari arsitektur ini adalah mudah digunakan dan dikelola, serta dapat diakses dengan mudah oleh berbagai aplikasi.

Keuntungan arsitektur berbasis file adalah:

1. Mudah digunakan dan dikelola. Karena data disimpan dalam bentuk file yang dapat dibuka dan diedit dengan mudah menggunakan aplikasi yang tersedia di komputer, maka proses pengelolaan data menjadi lebih sederhana.
2. Dapat diakses oleh berbagai aplikasi. File dapat diakses oleh berbagai aplikasi yang mendukung format file tersebut, sehingga data dapat digunakan oleh berbagai aplikasi yang berbeda.
3. Biaya rendah. Arsitektur berbasis file memerlukan biaya yang lebih rendah dibandingkan arsitektur berbasis database, karena tidak memerlukan software DBMS yang mahal.

Kelemahan arsitektur berbasis file adalah:

1. Tidak efisien dalam pengelolaan data. Karena data disimpan dalam file yang terpisah, maka proses pencarian dan pengelolaan data akan menjadi lebih rumit dan tidak efisien dibandingkan dengan arsitektur berbasis database.
2. Data tidak aman. Karena file dapat dibuka dan diedit oleh siapa saja yang memiliki akses ke file tersebut, maka data tidak aman dari risiko kehilangan atau pengubahan oleh pihak yang tidak berwenang.
3. Proses backup dan recovery data lebih sulit. Karena data disimpan dalam file yang terpisah, maka proses backup dan recovery data akan menjadi lebih sulit dibandingkan dengan arsitektur berbasis database.

Arsitektur berbasis database

Dalam arsitektur ini, data disimpan dalam sebuah database yang terpisah dari aplikasi yang menggunakannya. Aplikasi mengakses data melalui perintah-perintah yang ditentukan oleh sistem manajemen basis data (DBMS). Keuntungan dari arsitektur ini adalah data lebih aman dan mudah dikelola, serta dapat digunakan secara efisien dalam aplikasi yang berbeda.

Keuntungan arsitektur berbasis database adalah:

1. Efisien dalam pengelolaan data. Karena data disimpan dalam sebuah database yang terpisah dari aplikasi, maka proses pencarian dan pengelolaan data menjadi lebih efisien dibandingkan dengan arsitektur berbasis file.
2. Data aman. Karena data disimpan dalam sebuah database yang dikelola oleh DBMS, maka data lebih aman dari risiko kehilangan atau pengubahan oleh pihak yang tidak berwenang.
3. Proses backup dan recovery data lebih mudah. Karena data disimpan dalam sebuah database yang dikelola oleh DBMS, maka proses backup dan recovery data menjadi lebih mudah dibandingkan dengan arsitektur berbasis file.

Kelemahan arsitektur berbasis database adalah:

1. Mahal. Arsitektur berbasis database memerlukan biaya yang lebih tinggi dibandingkan arsitektur berbasis file, karena perlu menggunakan software DBMS yang mahal.
2. Sulit digunakan dan dikelola. Karena data disimpan dalam sebuah database yang dikelola oleh DBMS, maka proses pengelolaan data menjadi lebih sulit dibandingkan dengan arsitektur berbasis file.
3. Hanya dapat diakses oleh aplikasi yang kompatibel dengan DBMS. Karena data disimpan dalam sebuah database yang dikelola oleh DBMS, maka hanya aplikasi yang kompatibel dengan DBMS yang dapat mengakses data tersebut.

Jelaskan secara piramida level system keamanan!

Piramida sistem keamanan adalah model yang digunakan untuk menyusun tingkatan keamanan dari sistem yang paling dasar sampai yang paling tinggi. Setiap tingkatan menawarkan tingkat perlindungan yang lebih tinggi dari tingkat sebelumnya. Tingkatan-tingkatan dari piramida sistem keamanan biasanya meliputi:

1. Tingkat dasar: Ini merupakan tingkat pertama dari piramida sistem keamanan dan mencakup tindakan-tindakan dasar yang dilakukan untuk mengamankan sistem, seperti pengaturan sandi dan pembaruan perangkat lunak.
2. Tingkat fisik: Tingkat ini mencakup tindakan-tindakan yang dilakukan untuk mengamankan fisik dari sistem, seperti pembatasan akses fisik ke perangkat keras dan pengaturan kontrol akses.
3. Tingkat aplikasi: Tingkat ini mencakup tindakan-tindakan yang dilakukan untuk mengamankan aplikasi yang digunakan dalam sistem, seperti pembaruan aplikasi dan pengaturan hak akses.
4. Tingkat jaringan: Tingkat ini mencakup tindakan-tindakan yang dilakukan untuk mengamankan jaringan yang digunakan oleh sistem, seperti pengaturan firewall dan enkripsi jaringan.
5. Tingkat data: Tingkat ini mencakup tindakan-tindakan yang dilakukan untuk mengamankan data yang disimpan dalam sistem, seperti enkripsi data dan pembatasan akses data.

Terdapat 3 teknik pemrograman, sebutkan dan jelaskan!

Ada berbagai macam teknik pemrograman yang dapat digunakan, berikut tiga teknik pemrograman yang paling umum digunakan:

Pemrograman Berorientasi Objek (Object-Oriented Programming, OOP)

Teknik ini menekankan pada pembuatan objek yang mewakili konsep di dunia nyata dan interaksi antara objek-objek tersebut. OOP menggunakan konsep-konsep seperti kelas, objek, pewarisan, dan pemodelan polimorfisme untuk menyederhanakan proses pembuatan aplikasi yang kompleks.

Keuntungan Pemrograman Berorientasi Objek (OOP) adalah:

1. Abstraksi: OOP memungkinkan untuk menyembunyikan detail implementasi dan hanya menampilkan interface yang diperlukan.
2. Inheritance: OOP memungkinkan untuk menurunkan kelas dari kelas yang sudah ada, sehingga membuat kode lebih efisien dan mudah dikelola.
3. Polymorphism: OOP memungkinkan untuk menggunakan objek dengan tipe yang sama dengan cara yang berbeda-beda.
4. Encapsulation: OOP memungkinkan untuk menyembunyikan detail implementasi dari objek dan hanya menyediakan interface yang diperlukan.

Kelemahan Pemrograman Berorientasi Objek (OOP) adalah:

1. Overhead: OOP memerlukan lebih banyak kode dan waktu untuk menulis dan memahami program.
2. Kode yang lebih kompleks: OOP mengharuskan pengembang untuk memikirkan interaksi antar objek yang dapat menyebabkan kode yang lebih kompleks.
3. Performa: OOP dapat menurunkan performa karena adanya overhead yang disebutkan di atas.
4. Over-engineering: OOP dapat menyebabkan pengembangan yang berlebihan jika digunakan untuk masalah yang tidak sesuai.

Pemrograman Berorientasi Proses (Procedural Programming)

Teknik ini menekankan pada pembuatan sekumpulan prosedur atau fungsi yang dapat digunakan untuk menyelesaikan tugas-tugas tertentu. Pemrograman berorientasi proses menggunakan konsep-konsep seperti variabel, pengontrol aliran, dan subroutine untuk menyederhanakan proses pembuatan aplikasi.

Keuntungan Pemrograman Berorientasi Proses (Procedural Programming) adalah:

1. Sederhana: Prosedural programming lebih mudah dipahami dan dikelola dibandingkan dengan OOP.
2. Performa: Prosedural programming dapat lebih cepat dibandingkan dengan OOP karena tidak ada overhead dari objek dan kelas.
3. Scalability: Prosedural programming lebih mudah di-scalable dibandingkan dengan OOP karena tidak ada interaksi antar objek yang harus diperhatikan.

Kelemahan Pemrograman Berorientasi Proses (Procedural Programming) adalah:

1. Kode yang tidak terstruktur: Prosedural programming dapat menyebabkan kode yang tidak terstruktur dan sulit dikelola jika digunakan untuk masalah yang kompleks.
2. Maintainability: Prosedural programming dapat sulit di-maintain jika digunakan untuk masalah yang kompleks karena kode yang tidak terstruktur.
3. Reusability: Prosedural programming sulit digunakan kembali karena tidak adanya abstraksi dari objek dan kelas.
4. Lack of encapsulation: Prosedural programming tidak memiliki mekanisme untuk menyembunyikan detail implementasi dari objek.

Pemrograman Fungsional (Functional Programming)

Teknik ini menekankan pada pembuatan fungsi yang dapat digunakan untuk menyelesaikan tugas-tugas tertentu. Pemrograman fungsional menggunakan konsep-konsep seperti fungsi yang dapat digunakan sebagai argumen dan fungsi yang dapat mengembalikan nilai lain daripada nilai yang diharapkan.

Keuntungan Pemrograman Fungsional (Functional Programming) adalah:

1. Determinisme: Fungsional programming menjamin bahwa jika input sama, maka output juga sama, sehingga membuat program lebih mudah diprediksi dan di-debug.
2. Reusability: Fungsional programming menyediakan abstraksi yang dapat digunakan kembali dengan input yang berbeda.
3. Parallelism: Fungsional programming memungkinkan untuk mengeksekusi fungsi secara paralel karena tidak adanya efek samping dari perubahan data.
4. Pure functions: Fungsional programming menyediakan pure function yang tidak memiliki efek samping, sehingga membuat program lebih mudah diprediksi dan di-debug.

Kelemahan Pemrograman Fungsional (Functional Programming) adalah:

1. Overhead: Fungsional programming memerlukan lebih banyak kode dan waktu untuk menulis dan memahami program.
2. Kode yang lebih kompleks: Fungsional programming mengharuskan pengembang untuk memikirkan interaksi antar fungsi yang dapat menyebabkan kode yang lebih kompleks.
3. Performa: Fungsional programming dapat menurunkan performa karena adanya overhead yang disebutkan di atas.
4. Over-engineering: Fungsional programming dapat menyebabkan pengembangan yang berlebihan jika digunakan untuk masalah yang tidak sesuai.
5. Lack of mutability: Fungsional programming tidak memiliki mekanisme untuk mengubah data secara langsung, sehingga pengembang harus menggunakan alternatif seperti immutable data structures atau monads.

Jelaskan menurut Anda mengenai Trend Future Operation System!

Trend Future Operating System terutama difokuskan pada keamanan, kinerja, konektivitas, dan kemudahan penggunaan. Beberapa trend yang diharapkan akan muncul di masa depan adalah:

1. Sistem operasi berbasis cloud: Sistem operasi yang memungkinkan pengguna untuk mengakses aplikasi dan data dari mana saja melalui koneksi internet.
2. Sistem operasi yang lebih aman: Dengan semakin banyaknya serangan cyber, sistem operasi di masa depan akan dikembangkan dengan keamanan yang lebih baik untuk melindungi data pengguna.
3. Sistem operasi yang lebih cepat dan responsif: Peningkatan kinerja akan menjadi prioritas utama untuk meningkatkan pengalaman pengguna.
4. Integrasi dengan perangkat IoT: Sistem operasi di masa depan akan dapat terintegrasi dengan perangkat Internet of Things (IoT) untuk meningkatkan efisiensi dan produktivitas.
5. Sistem operasi yang lebih mudah digunakan: Dengan semakin banyaknya teknologi baru yang dikembangkan, sistem operasi di masa depan akan dibuat lebih mudah digunakan agar dapat digunakan oleh semua orang.
6. Sistem operasi yang lebih ramah lingkungan: Sistem operasi di masa depan akan dikembangkan dengan lebih ramah lingkungan untuk mengurangi dampak lingkungan dari teknologi.
7. Sistem operasi yang lebih personalisasi: Sistem operasi di masa depan akan dikembangkan untuk lebih personalisasi, sehingga dapat memberikan pengalaman yang lebih baik bagi pengguna.