

Білет №23

1. **Умова переходу зі стану P1 у P2 й навпаки**
Переходить з P2 в P1 якщо процеси знаходяться в одній черзі
2. **Які операції виконує програма першого рівня планування.** В інтерпретаторе настройка адресных констант происходит в процессе выполнения
3. **Вчому перевага алгоритмів покрокового конструювання.** Пошаговое конструирование – разделение всего процесса решения на определенное кол-во шагов и нахождение оптимального или квазиоптимального решения.
4. **Чому для оцінки придатності алгоритму використовують тимчасову складність, а не обчислювальну.** Временная сложность алгоритма - показывает время решения задачи в зависимости от размерности задачи. Если временная сложность алгоритма имеет экспоненциальный вид то алгоритм NP - полный. Наиболее приемлемым критерием является временная сложность алгоритма, позволяющая оценить изменение времени решения задачи от размерности, что существенно для параллельных систем. Вычислительная сложность, основанная на абстрактных понятиях ресурсов и времени, не дает столь точную оценку.
5. **Як визначити об'єм пам'яті, необхідної для завантаження програми динамічної-паралельної структури.** Корневой сегмент + наибольший суммирующий объем модулей, которые могут выполняться параллельно. Какая системная программа подготавливает информацию для работы загрузчика. Компилятор. (###редактор связей) Непосредственно связывающие – инфу – для них готовит компилер. (в спец. виде.)
6. **Назвати стратегії, що застосовуються в алгоритмах оптимізації.** При разработке стратегии решения данной задачи применяют следующие группы критериев оптимизации:

Критерии оптимизации относящиеся к вычислительной системе: процесс вычисления необходимо планировать так, чтобы обработка потока задач была оптимальна по эффективности использования ресурсов (процессоров, памяти, каналов связи), времени нахождения задач в системе, времени завершения задач, времени, затрачиваемого на выполнение пересылок, и т.д.
Критерии оптимизации самого процесса планирования: качество планирования, время решения задачи планирования или динамичность планирования в зависимости от того, что в данной ВС является критичным.
7. **Чому і коли блокується система переривань.** При дешифрации текущего прерывания
8. **Основна ідея «угорського алгоритму».** Венгерский алгоритм. Основная идея (википедия):
-если из всех элементов некоей строки или столбца вычесть одно и то же число , общая стоимость уменьшится на , а оптимальное решение не изменится;

-если есть решение нулевой стоимости, оно оптимально.

Алгоритм ищет значения, которые надо вычесть из всех элементов каждой строки и каждого столбца (разные для разных строк и столбцов), такие что все элементы матрицы останутся неотрицательными, но появится нулевое решение.
9. **Як змінюється вихідна інформація системи динамічного планування для однорідних і неоднорідних ОС.**
10. **Ідея розв'язання задач планування методом «раннього» планування**
Принцип раннего планирования (с конспекта): планирование выполняется как только есть все необходимые условия.

11. Що називається ядром супервізора, його функції. СУПЕРВІЗОР (supervisor).

Управляюча резидентна програма в складі операційної системи, координуюча розподіл і використання ресурсів вичислювальної системи. В операційній системі може бути декілька С. Наприклад, С. вводу-виводу контролює стан засобів вводу-виводу, ініціює і припиняє процеси вводу-виводу. С. основної пам'яті здійснює учет і динамічне розподілення області оперативної пам'яті, в яку завантажуються програми користувачів і деякі системні програми. С. організовує сторінковий обмін віртуальною пам'яттю. Нарешті, С. переривань аналізує переривання і вибирає необхідний вид обробки переривань. Розподіл апаратних і програмних ресурсів системи ставить складні завдання управління перед СУПЕРВІЗОРОМ, які він вирішує завдяки наявності спеціальних алгоритмів розподілення ресурсів системи

12. Що таке віртуальна операційна система. Коли вона застосовується. Віртуальна ОС -
Так звана ОС, яка дозволяє багатьом користувачам працювати на одній і тій же технічній базі (одно уст-во) одночасно працювати в різних операційних середовищах. Віртуальні ОС – дозволяють декільком користувачам працювати на одній і тій же машині на різних ОС.

13. Дати визначення термінам «завдання» й «програма», дані з точки зору ОС

Завдання – зовнішня одиниця роботи системи (описується на спеціальному мові). Завдання завантажується в систему тільки тоді, коли система має вільні ресурси і перетворюється в *задачу*.

Задача – внутрішня одиниця роботи системи, для якої система виділила ресурси крім процесорного часу. Задача фіксується в системі якщо їй виділені системні ресурси (блок управління процесом).

Процес – траєкторія процесора в адресному просторі. С точки зору ОС програми і дані – ресурс, виділений ОС.

Програма – ресурс задачі. В БУП – посилання на початок програми.

Дані – ресурс, оброблюваний задачею.

Види програм:

1. Повторно виконувані
2. Повторно не виконувані
3. Реінтерпретовані (т.н. pure procedures) для кожного входу обробляється свій блок даних, тіло процедури відокремлено від даних.

За структурою програми діляться на :

1. Прості структури
2. Оверлейні (т.н. плануваного перекриття)
3. Динамічно-последовательні

4. Динамически-параллельные

14. **Щотакеквазі-оптимальнерішення.** Квазиоптимизация - Метод решения задач оптимизации, в котором ищется не единственное наилучшее решение, а некоторое их множество, близких к нему.

Типичным примером задачи квазиоптимизации является обучение нейронной сети, которое производится не до достижения глобального минимума функции ошибки, а до некоторого достаточно малого ее значения. Такое решение называется квазиоптимальным.

15. **Якпідвищити ефективність роботи ОСзі спільною пам'яттю.** В шинно-ориентированных сетях процессоры соединяются с общей памятью посредством единого канала данных, называемого шиной. Такая система очень проста для реализации. Рис. 1.17(а) иллюстрирует обычную шинную архитектуру. При обращении процессора к общей памяти на определенных линиях шины выставляются соответствующие сигналы.

Главным недостатком данного способа является то, что при увеличении числа процессоров, связанных с общей шиной, весьма значительно увеличивается время ожидания каждым процессором доступа к памяти. Данный способ пригоден для небольшого числа процессоров.

Один из способов решения данной проблемы заключается в обеспечении каждого процессора локальной кэш-памятью, как показано на рис.1.17(б).

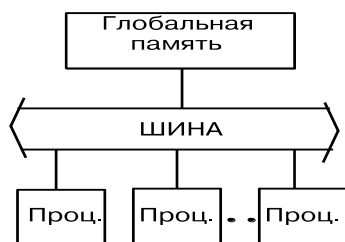


Рис.1.17.а

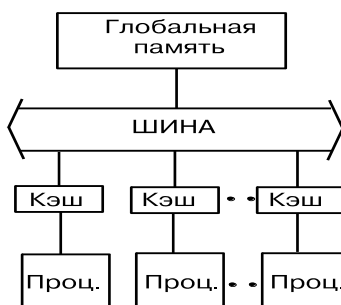


Рис. 1.17.б

Шинно-ориентированные сети Кэш шинно-ориентированные сети

Использование памяти в шинно-ориентированных сетях с FLTM топологией отличается от FITM BC. При обычных вычислениях, когда ссылка делается на какой либо фрагмент памяти, следующая ссылка вероятней всего будет сделана на фрагмент памяти, следующий за этим фрагментом. Этот вывод сделан на основе свойств "временной" и "пространственной" локальности теории рабочих множеств. Следовательно, весь этот фрагмент может быть помещен в локальную кэш-память процессорного элемента BC и всю дальнейшую обработку данных он может вести не обращаясь к общей памяти. При этом существенно сокращается время работы процессора с общей памятью. В случае кэш-промаха (т.е. когда в

локальной памяти не оказалось требуемого слова) процессору необходимо еще раз обратиться к общей памяти.