

Білет 3.

1. **Недоліки методу колективного доступу.** Смуга пропускання не належить одному користувачу, можуть бути проблеми з QoS.
2. **Особливості розподілених операційних систем.** Розподілена ОС – сукупність обчислювальних вузлів, з'єднаних між собою каналами зв'язку, з точки зору користувача представляє собою єдине ціле. Відсутність спільної пам'яті призводить до неможливості визначення загального стану за допомогою множини спільних змінних, а неможливість спільного звернення до пам'яті та різниця в затримках передач повідомлень призводить до того, що при визначенні стану будь-якого елементу системи з різних точок можна отримати різні результати. Виконання роботи розподіляється у вузлах, виходячи з міркувань пропускну здатності усієї системи. Розподіленні системи мають високий рівень організації паралельних обчислень.
3. **Структура модуля** Специфікація та тіло.
4. **Стратегії, що використовуються при динамічному плануванні.** Пошук максимального паросполучення, визначення конфліктних призначень, виділення підматриць.
5. **Що таке транзитні програми ОС?** Які завантажуються у пам'ять тільки при необхідності.
6. **Резиденція системи. Які програми там знаходяться?** Це ОС, що отримується в результаті генерації. В ній знаходяться резидентні програми, тобто програми, що завжди в пам'яті.
7. **Програма динамічної послідовної структури. Чим вона краще інших структур.** Програми, налаштовані за допомогою модульного принципу, представлені в переміщувальному вигляді, можуть підгружатися за мірою необхідності з організацією зв'язків по керуванню та даними. Краще інших, тому що пам'ять виділяється тільки коли потрібна.
8. **Коли застосовують евристичні алгоритми?** Коли висока обчислювальна складність.
9. **Дати визначення балансового планування. Які проблеми в його вирішенні?** Система планує навантаження, коли задача не вирішується – зовнішній планувальник намагається визначити чи правильне розподілення на вузлах. Проблеми – як визначити, що програма вісить? Передбачити звільнення, проблема міграції.
10. **Що таке резидентні вершини? Як їх знайти та як можна їх використати?** Вершини, що увійшли в критичний шлях, а їх часові та просторові координати не повинні змінюватись при вирішенні задач. Знайти критичний шлях.
11. **Недоліки використання мультипрограмування.** Може збільшитися час вирішення окремої задачі, завантаження процесорів збільшується, якщо задача захватить процесор, то вона його не віддає.
12. **Основна відмінність використання підпрограм в компілюючих системах и інтерпретуючих.** Інтерпретуючі переводять частину програми в машинні команди і одразу виконують, виконуючи налаштування адресних констант. Компілючі створюють об'єктний модуль, який потім обробляється редактором зв'язків.
13. **Як знайти максимально й мінімально необхідну кількість процесорів?** Максимальна ширина ярусу ярусно-паралельної форми – максимальна кількість. Мінімальна:

$$N_{\min} = \left\lceil \frac{\sum_{i=1}^n t_i}{T_{kp}} \right\rceil, N_{\text{low}} = \lceil \text{Sum}(T_i) / T_{kp} \rceil$$

14. **Ідея застосування алгоритмів кластеризації. Шляхи визначення вузлів, що кластеризуються.** Зменшення графу. Знаходження критичної вершини, що визначає час вирішення.
15. **Навіщо потрібен рівень реконфігурації в системі планування багатопроцесорної паралельної системи?** Для продовження роботи в разі виходу з ладу якогось обладнання.