

Білет 10.

1. **Який механізм являється основою багатопрограмного режиму роботи?** Механізм переривань
2. **Умова переходу процесу з активного стану у готовий або підготовлений?** Коли процес довго виконується та двигается достатньо далеко, то планувальник процесів вирішує, що процесор необхідно віддати іншому процесу, після чого переводить цей процес в стан підготовка та віддає процесор іншому. Коли ресурс виділяється процесу, то він переходить в стан готовий.
3. **Властивості модуля. Пояснити необхідність кожного.** Властивості: стандартна внутрішня структура, взаємна незалежність модулів, функціональна завершеність, параметрична універсальність. Вихідний модуль містить в собі код модуля, який потім буде виконуватись. Об'єктний модуль містить в собі скомпільований код, який потім може бути об'єднаний з іншими модулями для виконавчого модуля або бібліотеки. Завантажувальний модуль готовий до виконання, але в ньому не налаштовані адресні константи, написаний на машино-орієнтованій мові. Абсолютний модуль є виконавчим, в ньому визначені адресні константи.
4. **Причини виникнення «безкінечного» відкладання. Чому воно небезпечне?** Безкінечне відкладання може виникнути якщо процес має малий пріоритет коли планувальник розподіляє ресурси по пріоритетах. Постійно будуть приходити більш пріоритетні процеси, а цей буде безкінечно довго очікувати виділення ресурсів. Це небезпечно тим, що безкінечне очікування може призвести до тупика, який важко відловити.
5. **Дати визначення транзитного модуля. За якою схемою завантажуються транзитні модулі і в яку область пам'яті?** Транзитний модуль – модуль, який завантажуються в пам'ять тільки тоді, коли він потрібен. Завантажуються по оверлейній або динамічно-послідовній схемі в транзитну область. Але зараз завантажуються в будь-яку область ОП.
6. **Які програми знаходяться у ядрі операційної системи? (Види програм)** Програми, що оброблюють переривання, контролюють процеси, забезпечують їх взаємодію, підтримують операції вводу/виводу, підтримують роботу файлової системи, диспетчеризація процесів, задач, ресурсів, ведення обліку роботи машини.
7. **Яку інформацію та як компілятор передає завантажнику, що налаштовує?** Компілятор передає інформацію про адресні константи в даній команді за допомогою бітів перемістимості.
8. **Чому NP-повні задачі не придатні для динамічного планування?** Тому що вони не можуть бути вирішені за поліноміальний час n^k . А при динамічному плануванні основна ціль – зменшити час планування, тому динамічні алгоритми мають як правило лінійну складність.
9. **Чому й коли блокується система переривань?** При дешифрації поточного переривання, тому що дешифрація є атомарною операцією.
10. **Які методи використовуються для розв'язання NP-повних задач?** Евристичні алгоритми, псевдополіноміальні, метод локальних покращень, метод гілок та границь, метод випадкового пошуку.

11. **Що таке «спулінг», чому він схильний до тупикових ситуацій?** Спулінг – на вході в систему обслуговування одна швидкість, на виході інша. Задачам при цьому створюється ілюзія одночасного доступу до пристроїв, але задачі працюють без прямого доступу до пристроїв. Спулінг допомагає уникнути тупиків, оскільки напряму з пристроєм працює лише демон.
12. **Дати визначення розподіленій системі.** Розподілена ОС – сукупність обчислювальних вузлів, з'єднаних між собою каналами зв'язку, з точки зору користувача представляє собою єдине ціле.
13. **Що таке базове рішення та у чому ідея його оптимізації?** Базове рішення – перше правильне рішення задачі. Оптимізація його зменшує час вирішення задачі.
14. **Навіщо потрібно виділяти обов'язкові призначення?** Для того, щоб знайти максимальне паросполучення та щоб можна було виявити конфліктні призначення.
15. **Як визначити граничну кількість процесорів для завантаження програм у системі з загальною пам'яттю?** Максимальна ширина ярусу ярусно-паралельної форми –

максимальна кількість. Мінімальна:

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