

## Білет №6

### 1. Дати визначення програми з точки зору операційної системи.

Ничего адекватного пока не нашел

### 2. Умова переходу процесу з активного стану в підготований



1. Процесс блокируется, ожидая входных данных
2. Планировщик выбирает другой процесс
3. Планировщик выбирает этот процесс
4. Доступны входные данные

**Рис. 2.2.** Процесс может находиться в состоянии выполнения, готовности или блокировки. Стрелками показаны возможные переходы между состояниями

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.

### 3. Зв'язок модулів за керуванням. Які операції виконуються й якими програмами? Яка інформація й куди при цьому записується?

Связь по управлению предусматривает сохранение состояния модуля на момент перехода на вызываемый модуль, причем действия по этому сохранению выполняет вызываемый модуль. Операции: возврат из  $i-1$  модуля только в  $i+1$ ; возврат из  $i+1$  в  $i$ . Вызывающий модуль после передачи управления стирается. Виды программ могут быть: повторно не исполняемые, повторно исполняемые, чистые процедуры. (via common regs) Связь по управлению – связь для организации возврата. Возврат модуль В должен иметь команду, делающую так, что в области сохранения модуля А сохраняется состояние регистров. В модуле В есть адрес области сохранения в модуле А. Сохраняется адрес возврата, для возврата адреса следующей команды.

### 4. Умови переходу з P3 у P1 або P2.

Выход из прерывания, вызов сист. Функции из прерывания

P4 - машина не работает, выхода из него нет

P3 - состояние, в которое переходит если зафиксировано прерывание. атомарная операция расшифровки прерывания

P2 - программы пользователя (прикладные)

P1 - система обрабатывает системные процессы

**5. При наявності віяла, яке ребро береться у рішення? Любые две вершины, остальные редуцируются.**

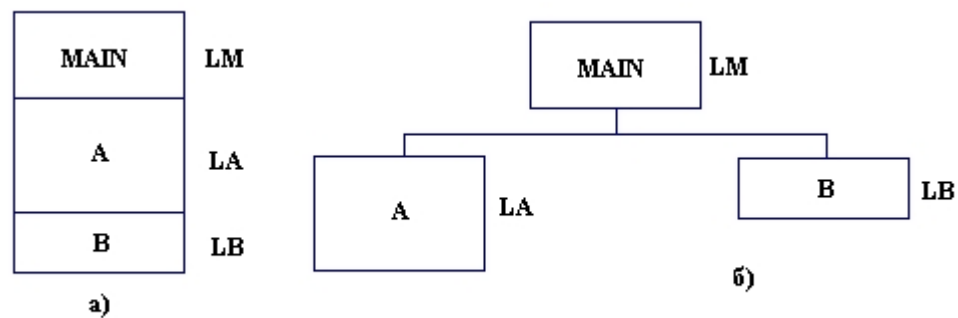
**6. Основні дії програми початкового завантаження. Де вона зберігається та як її знайти?**

POST(проверка оборудования). Загрузка бутового загрузчика и передача ему управления. Лежит в BIOS.

**7. Як визначити об'єм пам'яті, необхідної для завантаження програми оверлейної структури?**

**Теория:**

Оверлей - это такой способ использования оперативной памяти, при котором в один и тот же участок памяти, называемый оверлейным буфером, попеременно по мере надобности загружаются различные оверлейные (перекрывающиеся) модули. При этом все оверлейные модули в готовом к работе виде хранятся на диске, а в оперативной памяти в каждый момент находится лишь один активный модуль и, возможно, небольшое число неактивных.



**Рис.11.1.** Пример структуры программы: а) неоверлейная; б) оверлейная

**Ответ:**

Размер памяти = размер главного модуля + максимальный размер оверлейного модуля. Если модули вызывают друг-друга, то максимальная сумма одновременно загруженных модулей

**8. Дати визначення ГРІД системи.**

Совокупность машин для решения задач большой сложности

**9. Як визначити зону знаходження оптимального розкладу для систем зі спільною пам'яттю?**

**10. Що таке строго неоднорідна ОС? Яка задача має розв'язуватися при плануванні?**

Заявка не може бути виконана на даному ресурсі.

**Проблемы:**

1. Optimal performance (Производительность)

2. Core assignment balance(Баланс нагрузки)
3. Response time fairness.

### 11. Що шукає алгоритм Мальгранжа та який його основний недолік?

Ищет нулевую подматрицу для матрицы отображения двудольного графа, чтобы можно было обнулить симметричную ей подматрицу. Ищутся суммы единиц в строках и столбиках.

разбиение графов на сильно связные под-графы.

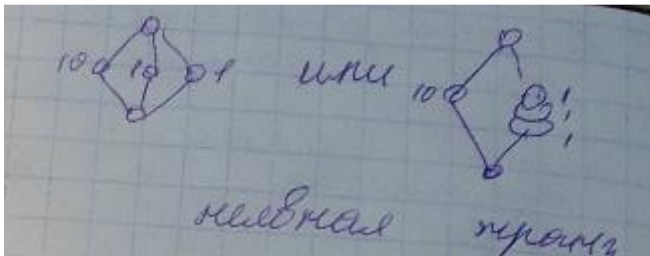
### 12. Яку задачу розв'язує лінійний планувальник? Яка стратегія при цьому використовується?

### 13. Чим відрізняється матриця зв'язності від матриці інцидентності?

Связность показывает связь между  $i$ -й и  $j$ -й вершинами

Инцидентность показывает соседство вершин и ребер(строка – вершина, столбец – ребро)

### 14. Дати приклад неявної транзитивності .



### 15. Засоби оцінки складності алгоритму.

Временная сложность – зависимость времени решения задачи от её размерности.

Вычислительная – время решения зависит от количества операций, которые выполняются для решения. Ёмкостная – какое оборудование нужно для решения задачи.