

Операционные
Системы

Практики

1

1° Перечислите основные ф-ии программы-диспетчеров —
— преимущественников операционных систем на этапе однопрограммной пакетной обработки

Ответ:

- повторное использование кода
- автоматизация линковки загрузки

2° Перечислите основные ф-ии операционных систем, появившихся на этапе мультипрограммной обработки.

Ответ:

- разделение процессорного времени
- разделение памяти
- защита памяти
- планирование комплексного использования ресурсов
- универсальный доступ к информации на внешних устройствах
- коммуникация между программами

3° Системный вызов — обращение пользовательской проги к О.С. с „просьбой” предоставить ресурс или выполнить привилегированную операцию

4° Прерывание — сигнал, поступающий от внешнего устройства к СРЦ, сообщаящий о наступлении опр. события, приостанавливающий текущий поток выполнения команд и передающий управление подпрограмме обработчик прерываний

5° Виртуальная память — это абстракция, позволяющая при создании программы использовать адреса, отсчитываемые от виртуального 0, а при исполнении заменять на физические

6° Операционная система — это базовое системное программное обеспечение, управляющее работой вычислительного узла и реализующее универсальный интерфейс между аппаратным обеспечением, программным обеспечением и пользователями

2

1° Основные цели работы современной О.С.

- обеспечить производительность, надежность, безопасность исполнения пользовательского П.О., хранение и доступа к данным, диалога с пользователем и использование программного обеспечения.
- управление разработкой и использованием П.О.
- оптимизация использования ресурсов
- поддержка использования вычислительного узла
- поддержка развития О.С. ?

2° Задачи, которые решает О.С.

в рамках организации интерфейса между пользовательскими приложениями и аппаратным обеспечением П.к.

- API
- управление исполнением программы
- обнаружение и обработка ошибок
- высокоуровневый доступ к устройствам I/O
- управление хранилищем данных
- мониторинг ресурсов

3° Основной принцип обеспечения О.С. эффективного использования ресурсов (PDSA) - Plan - do - check - act -

Решение многокритериальной задачи:

$$\begin{cases} K' = a_1 \cdot k_1 + \dots + a_n \cdot k_n & \text{- суперкритерий} \\ a_1 + \dots + a_n = 1 & \text{- весовые коэффициенты} \end{cases}$$

k_1, \dots, k_n - критерии для процессов
нужно найти, когда K' - макс

Plan - формирование критериев / выбор алгоритма планирования

Do - выполнение плана

Check - проверка предсказанных и полученных результатов, достижение целевых показателей

Act - решение проблемы планирования

4° Какие основные механизмы используются подсистемой управления процессами

- дискрипторы процессов
- планировщик

5° Какие основные механизмы используются подсистемой управления файлами

- преобразование имен
- каталоги

6° Два основных способа организации пользовательского интерфейса в О.С.

- CLI
- GUI

3

1° В чем отличие программного кода, исп. в режиме ядра от user?

В режиме ядра код исполняется в привилегированном режиме, может что-то ещё

2° Основные подходы, заложенные в основу арх. современных ОС

- Модульная архитектура
- функциональная избыточность
- функциональная избирательность
- параметрическая универсальность
- многоуровневая иерархическая вычислительная система

3° Слои в многослойной арх. ОС.

- Hard Ware ?
- Аппаратная поддержка ядра
- Машинно-зависимые модули
- Базовые механизмы ядра
- Менеджер ресурсов
- Системные вызовы / API
- Soft Ware

4° Средства аппаратной поддержки ядра — не уверено.

На уровне железа.

Назначение: Реализовать необходимые устройства для работы ОС.:

таймер, систему прерываний и т.д.

5° Функции ОС в микроядре

- Системные вызовы
- Базовые механизмы ядра
- Машинно-зависимые модули (HAL)

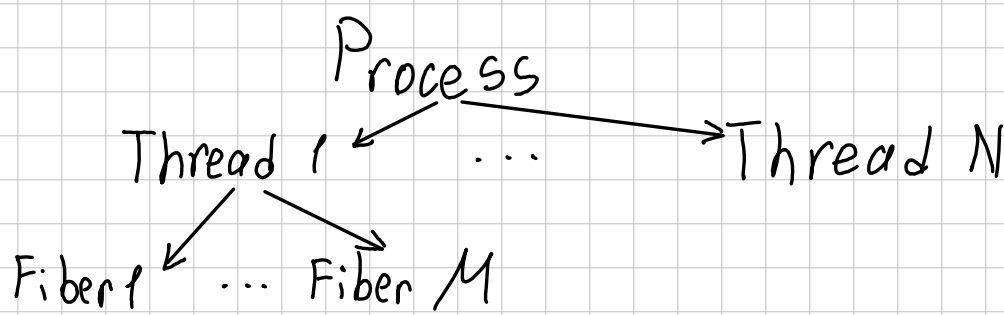
6° Многослойная монокриптная архитектура в сравнении с микроядерной.

+ : Производительность, меньшее потребление RAM (т.к. многие компоненты в самом ядре)

- : Плохо расширяема, плохо подходит для распределенных систем

4

1° Процесс, поток, волокно - отличия



Совокупность набора исполняемых команд и контекста исполнения = A

Процесс - A + ассоциированные с ним ресурсы под управлением О.С.

Поток - A , разделяющий все/часть ресурсов процесса, под управ. О.С.

Волокно - набор исп. команд в общем контексте потока. О.С. волокно не видит

2° Ф-ии подсистемы управ. процессами

- Создание процесса
- Обеспечение ресурсами
- Изоляция
- Планирование
- Диспетчеризация
- Межпроцессорное взаимодействие
- Синхронизация
- Завершение (уничтожение)

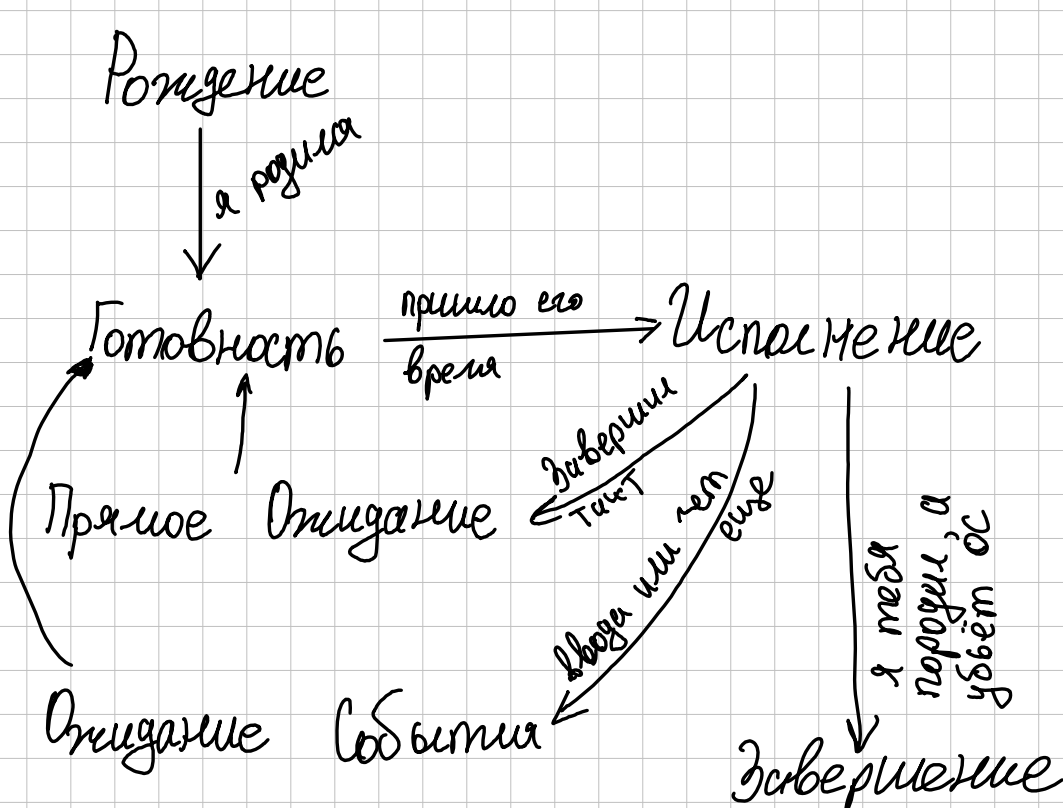
3° Действия ОС при диспетчеризации

Видимо смена регистров, вставка/удаление процесса из очереди. Переключение процессов между состояниями.

4° Информация в дескрипторе (PCB)

- Индикатор (PID, PPID, UID)
- Состояние (статус, контекст)
- История процесса

5° 5-ти уровневая модель жизни



1° Виде планирования в ОС и его место в жизненном цикле процесса (из какого состояния переводится)

- Краткосрочное: готовность \leftrightarrow исполнение

- Среднесрочное:
 - готовность \leftrightarrow готовность вне памяти
 - ожидание \leftrightarrow ожидание вне памяти

- Долгосрочное: ^{вне памяти} рождение \Leftrightarrow готовность

2° Критерии планирования процессов

- Качественные:

- справедливый

- эффективный

- Измеримый:

- полное время исполнения

- среднее время отклика

- сокращение времени отклика

3° Разница вытесняющих и невытесн.

- Невытесняющие - если процесс начал выполняться, он может прерваться только сам (сменить состояние)

- Вытесняющие - есть механизмы прерывания выполнения текущего процесса.

4° В каком смысле $FCFS \approx Round Robin$

При очень малом кванте, тогда идут
зоп. расходы на переключения (каждый токет)

5°

$P_0 - 0 - 2$

$p_2 - 0 - 7$

$$K=3$$
$$P_1 = 1 - 3$$

$P_3 = 12 - 4$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
p_0	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Ч	Ч	Ч	Г	Г	Г	Г	Ч	Ч	Ч	Ч	Ч		
p_1			Г	Г	Ч	Ч	Ч																	
p_2	Ч	Ч	Ч	Г	Г	Г	Ч	Ч	Ч	Ч														
p_3													Г	Ч	Ч	Ч	Ч							

6

1)
 1 shared int lock = 0;
 2 while(lock);
 3 lock = 1;
 4 <critical section>
 5 lock = 0;

Если прерывание ^(P₀) произойдет между строк 2, 3, то P₀ сможет войти в crit section, но и P₀ тоже сможет войти (если будет прерывание)

2) Строгое чередование гарантирует взаимное исключение (но не прогресс)
 P₀ завершит работу с R, передаст его P₁, P₁ в ожидании, P₀ снова начал выполняться, но не может получить R, т.к. он у P₁
 Целесообразно использовать, когда необходимо взаимное исключение.

3) Может возникнуть тупик, если прерывание произойдет между 2, 3.
 P₀ встанет в готовность → P₁ тоже же встанет → друг друга ждут

4) P(S) {
 while (s == 0);
 s--;
 }
 V(S) {
 s++;
 }

5) Th Коффмана:

Мн-во процессов окажется в тупике ⇔ выполняются условия:

- 1° Одновременно N ресурс может исп. один процесс
- 2° Процессы могут удерживая рес. ожидать следующий
- 3° Нельзя принудительно отобрать ресурс у процесса
- 4° ∃ конечная цель процессов