

Планирование процессов

Уровни планирования процессов:

**долгосрочное*

**краткосрочное*

**среднесрочное*

Критерии планирования и требования к алгоритмам:

- * Справедливость - гарантировать каждому заданию или процессу определенную часть времени использования процессора в компьютерной системе, стараясь не допустить возникновения ситуации, когда процесс одного пользователя постоянно занимает процессор, в то время как процесс другого пользователя фактически не начинал выполняться.
- * Эффективность - постараться занять процессор на все 100% рабочего времени, не позволяя ему простаивать в ожидании процессов, готовых к исполнению. В реальных вычислительных системах загрузка процессора колеблется от 40 до 90%.
- * Сокращение полного времени выполнения (*turnaround time*) - обеспечить минимальное время между стартом процесса или постановкой задания в очередь для загрузки и его завершением.
- * Сокращение времени ожидания (*waiting time*) - сократить время, которое проводят процессы в состоянии готовности и задания в очереди для загрузки.
- * Сокращение времени отклика (*response time*) - минимизировать время, которое требуется процессу в интерактивных системах для ответа на запрос пользователя.

Независимо от поставленных целей *планирования* желательно также, чтобы алгоритмы обладали следующими свойствами:

- * Были предсказуемыми. Одно и то же задание должно выполняться приблизительно за одно и то же время. Применение алгоритма *планирования* не должно приводить, к примеру, к извлечению квадратного корня из 4 за сотые доли секунды при одном запуске и за несколько суток - при втором запуске.
- * Были связаны с минимальными накладными расходами. Если на каждые 100 миллисекунд, выделенные процессу для использования процессора, будет приходиться 200 миллисекунд на определение того, какой именно процесс получит процессор в свое распоряжение, и на переключение контекста, то такой алгоритм, очевидно, применять не стоит.
- * Равномерно загружали ресурсы вычислительной системы, отдавая предпочтение тем процессам, которые будут занимать малоиспользуемые ресурсы.
- * Обладали масштабируемостью, т. е. не сразу теряли работоспособность при увеличении нагрузки. Например, рост количества процессов в системе в два раза не должен приводить к увеличению полного времени выполнения процессов на порядок.

Вытесняющее и невытесняющее планирование:

Планировщик может принимать решения о выборе для исполнения нового процесса из числа находящихся в состоянии готовности в следующих четырех случаях.

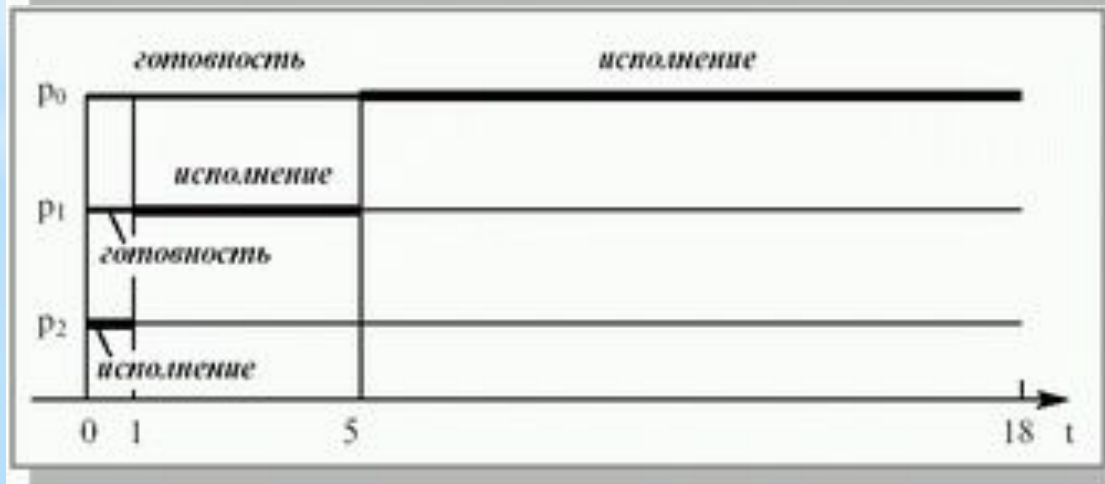
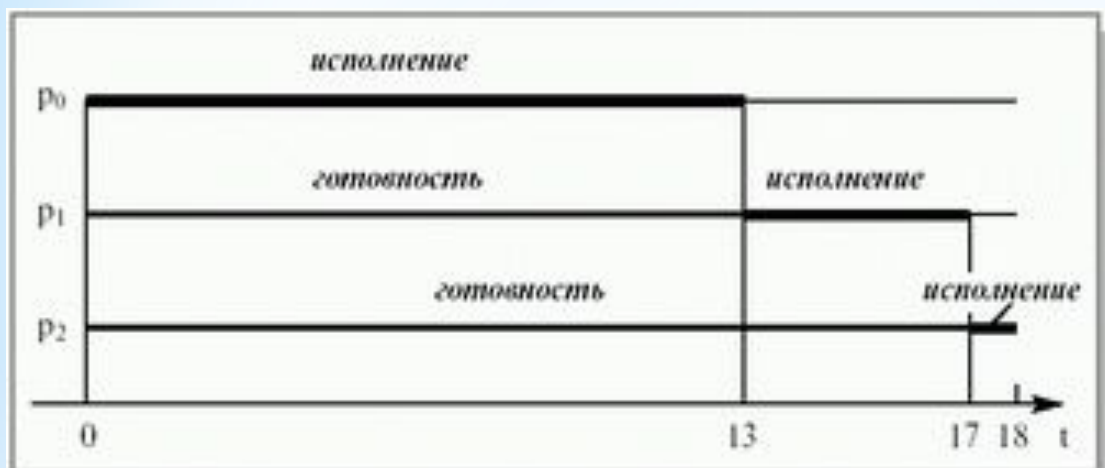
- * Когда процесс переводится из состояния исполнение в состояние закончил исполнение.
- * Когда процесс переводится из состояния исполнение в состояние ожидание.
- * Когда процесс переводится из состояния исполнение в состояние готовности (например, после прерывания от таймера).
- * Когда процесс переводится из состояния ожидание в состояние готовности (завершилась операция ввода-вывода или произошло другое событие).

Алгоритмы планирования

First-Come, First-Served (FCFS)

Таблица 3.1.

| Процесс | p_0 | p_1 | p_2 |
|--|-------|-------|-------|
| Продолжительность очередного CPU burst | 13 | 4 | 1 |



Shortest-Job-First (SJF)

Таблица 3.5.

| | | | | | | | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| Время | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| P ₀ | Г | Г | Г | Г | И | И | И | И | И | | | | | | | |
| P ₁ | Г | И | И | И | | | | | | | | | | | | |
| P ₂ | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| P ₃ | И | | | | | | | | | | | | | | | |

Таблица 3.4.

| | | | | |
|--|----------------|----------------|----------------|----------------|
| Процесс | P ₀ | P ₁ | P ₂ | P ₃ |
| Продолжительность очередного CPU burst | 5 | 3 | 7 | 1 |

Таблица 3.7.

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| В | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | |
| р | | | | | | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| р | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | | | | | | | |
| р | | | И | И | | | | | | | | | | | | | | | | |
| р | | | | | | | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| р | И | И | Г | Г | И | И | И | | | | | | | | | | | | | |
| р | | | | | | | | | | | | | | | | | | | | |

Таблица 3.6.

| Процесс | Время появления в очереди очередного CPU burst | Продолжительность |
|----------------|--|-------------------|
| P ₀ | 0 | 6 |
| P ₁ | 2 | 2 |
| P ₂ | 6 | 7 |
| P ₃ | 0 | 5 |

Многоуровневые очереди (Multilevel Queue)

