

Задача 1 — Обедающие философы

Кашин Андрей

23 марта 2013 г.

1 Алгоритм

Идея алгоритма: Когда философ хочет есть, он проверяет, доступны ли вилки слева и справа, при этом не блокируя. После этого он кричит, что сейчас он будет пытаться начинать есть, блокирует глобальный для всех философов mutex. Снова проверяет доступны ли вилки, и в случае доступности обеих берет их. После он разблокирует глобальный mutex. Если ему удалось взять две вилки, он начинает есть. Иначе он ждет некоторое время и повторяет попытку начать есть.

2 Теоретическое обоснование

Проверим свойства этого алгоритма. Очевидно, что deadlock'ов нет, так как вилки берутся парой, и если философ не может взять две вилки одновременно, он будет ждать. Таким образом если вилок нет на столе, то кто-то ест, а если они есть на столе, то первый кто их может взять точно возьмёт их и начнет есть. Ни один философ не будет голодать, так как время приёма пищи и время размышлений случайное. Тогда можно посмотреть на каждого конкретного философа, условием голодания будет то, что его соседи будут по очереди в состояниях (1 думает, 2 ест) и (1 ест, 2 думает), тогда этому философу не будет доставаться вилка. Но такая последовательность событий не может быть очень длинной, так как вероятность того, что отрезки размышлений соседей не пересекаются убывает экспоненциально. А так как все философы находятся в одинаковых условиях, этот эффект будет сглаживаться. Равномерное распределение ресурсов следует из тех же утверждений симметричности, и подтверждается результатами исполнения программы.

Одновременно могут(и будут) есть несколько философов. А именно, если есть n философов, и все они хотят есть, то $n/3$ из них будут гарантированно есть, так как прием пищи одного философа запрещает прием пищи максимум двум другим философам.

Так как количество принимающих пищу растет как $O(n)$, то решение является масштабируемым при увеличении количества философов.

3 Результаты тестовых запусков

Важным критерием является то, что при одинаковом максимальном времени размышлений и приема пищи, время ожидания философов в среднем составляет 28% от общего времени работы для любого количества философов и любого времени работы, а среднее количество приемов пищи составляет $69 * T$, где T — общее время работы. Это показывает, что алгоритм является масштабируемым, так как количество приемов пищи растет линейно от времени, а доля времени ожидания не изменяется.

Та же стабильность, только с другими коэффициентами наблюдается для других значений максимального времени размышления приема пищи. Таким образом эффективность алгоритма можно оценить некоторой формулой, зависящей только от отношения максимальных времен размышления и приема пищи, что символизирует идеальную масштабируемость.