

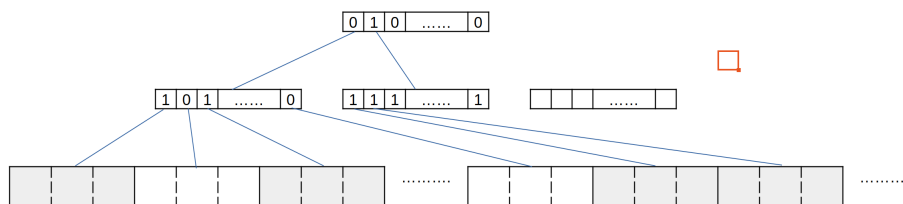
"Slab" Allocator

1 Введение

- Есть диапазон ячеек `TotalSize` (например, $8 \cdot 1024^2$). В этом диапазоне требуется уметь выделять непрерывные сегменты ячеек и позже освобождать их. При этом размер `Size` сегмента не более `MaxBufferSize`.
- Хотим это делать за $O(1)$.
- Весь диапазон ячеек разбивается на блоки размера `BlockSize` для выделения сегментов фиксированного размера.
- Должны выполняться следующие ограничения:
 - $\text{MaxBufferSize} \in [1, \dots, 64]$.
 - $\text{BlockSize} \% 64 = 0$.
 - $\text{BlockSize} \leq 64^2 = 4096$.
 - $\text{TotalSize} \% \text{BlockSize} = 0$.

2 Структура блока

- Каждый блок выделяет сегменты фиксированного размера: $\text{Size} \in [1, \dots, \text{MaxBufferSize}]$.
- Блок разбивается на сегменты, каждый сегмент непрерывная последовательность из `Size` ячеек, количество сегментов равно $\lfloor \frac{\text{BlockSize}}{\text{Size}} \rfloor$. В конце может остаться ($\leq \text{Size} - 1$) неиспользуемых ячеек.
- Сегменты разбиваются на группы по 64. Последняя группа может содержать меньше 64 сегментов. Максимальное количество групп равно 64 при $\text{BlockSize} = 4096$ и $\text{Size} = 1$.
- Для хранения информации о занятых сегментах используются битовые маски (64 битные числа): 0 - свободен, 1 - занят. Для каждой группы хранится своя битовая маска (64 штуки), также отдельная битовая маска `group_mask` хранит информацию о том, есть ли свободные сегменты в группе.



- Сложность выделения или освобождения одного сегмента в блоке – $O(1)$.
- Максимальное количество ячеек не попавших ни в один из сегментов (дыра в конце блока) равно 49 при размере блока 4096, т.е. примерно 1%.
- Размер метаданных одного блока равен 536 байт. Если одна ячейка соответствует 8 байт полезных данных, то получается размер метаданных примерно 1.6% от полезных данных.

3 Использование блоков

- Каждый блок может находиться в одном из статусов:
 1. Свободный
 2. Частично занят
 3. Полностью занят
- Возможные изменения статусов: $1 \leftrightarrow 2$, $2 \leftrightarrow 3$.
- В статусах "Частично занят" и "Полностью занят" блок хранит сегменты одной длины.
- Все блоки со статусом "Свободный" хранятся в общем списке (двухнаправленном).
- Для каждого допустимого размера сегмента $\text{Size} \in [1, \dots, \text{MaxBufferSize}]$ создается свой список блоков со статусом "Частично занят".
- Блоки при переходе в статус "Полностью занят" удаляются из списка блоков соответствующего размера, а при переходе обратно в статус "Частично занят" добавляются в начало списка.

4 Алгоритм выделения памяти

- Ищем блок в котором будет выделен сегмент:
 - Проверяем, если список частично занятых блоков этого размера не пустой, то используем первый блок.
 - Если список свободных блоков не пуст, то используем первый блок. Блок удаляется из этого списка и добавляется в список частично занятых блоков этого размера.
 - Если не удалось найти блок, то возвращаем ошибку.
- В блоке ищем первую доступную группу, в ней первый свободный сегмент. Возвращаем индекс первой ячейки сегмента:

$$\text{cell_index} = \text{block_index} \cdot \text{BlockSize} + \text{segment_index} \cdot \text{Size}$$

5 Алгоритм освобождения памяти

- При освобождении передается индекс первой ячейки и размер сегмента. Проверяем, что они не выходят за допустимые границы.
- По индексу ячейки определяем индекс блока и индекс сегмента:

$$\text{block_index} = \lfloor \frac{\text{cell_index}}{\text{BlockSize}} \rfloor$$
$$\text{segment_index} = \frac{\text{cell_index} - \text{BlockSize} \cdot \text{block_index}}{\text{Size}}$$

- Делаем следующие проверки:
 - В этом блоке хранятся сегменты нужной длины.
 - Индекс ячейки правильно восстанавливается по вычисленным значениям.
 - Сегмент отмечен как занятый.
- Отмечаем в блоке сегмент как свободный.
- Если блок стал пустым, переводим его в статус "Свободный". Если блок до этого "Полностью занят то переводим его в статус "Частично занят" и добавляем в соответствующий размеру список.