

MicroTESK - Task #4055

[mmu] Определение политики вытеснения PLRU

04/02/2013 08:26 AM - Alexander Kamkin

Status:	Closed	Start date:	04/02/2013
Priority:	Normal	Due date:	
Assignee:	Taya Sergeeva	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.10 hour
Detected in build:	svn	Published in build:	
Description			
Насколько я понимаю, политика вытеснения PLRU (Pseudo Least Recently Used) до сих пор не определена.			

History

#1 - 04/11/2013 06:16 PM - Alexander Kamkin

Я обещал описать алгоритм. Если я ничего не путаю, алгоритм следующий.

Пусть ассоциативность равна N.

Для вытеснения используются N бит (bits). В начальный момент они инициализируются нулем (bits = 0;).

При попадании в колонку с номером i, в i-ый бит записывается 1 (bits |= (1 << i);). Если все биты при этом становятся равными 1, они все дружно обнуляются 0 (if(bits == (1 << N) - 1) { bits = 0; }).

При промахе ищется индекс первого нулевого бита (for(int i = 0; ...) { if((bits & (1 << i)) != 0) { return i; } }).

#2 - 04/15/2013 12:46 PM - Taya Sergeeva

- Status changed from New to Resolved

r1178.

#3 - 04/20/2013 09:19 AM - Alexander Kamkin

- Status changed from Resolved to Open

Таня, нужно проверять то, что пишется. Я говорю даже не о тестах, а визуальной инспекции кода.

Для чего, например, нужен массив plru? Он заполняется в конструкторе, но потом не используется. Или, например, вот этот код:

```
if (bits != (1 << i))
{
    bits = 1;
}
```

У меня было написано bits |= (1 << i) (установка в 1 i-ого бита), а не bits != (1 << i).

Как в коде выражается условие попадания в i-ую строку множества?

Нельзя писать код, если до конца не понимаешь, как устроен алгоритм. Получается никому не нужный код, который остается только выкинуть и написать новый. Какая от этого польза? Зачем тратить на это время?

Подход простой:

1. Понять (в деталях) как устроен алгоритм/программа/система (какие функции, как эти функции реализуются и т.п.);
2. Аккуратно запрограммировать задуманное (каждая строчка должна быть осмысленной).
3. Не менее аккуратно верифицировать/протестировать написанный код.

P.S. Еще раз. Нужно понимать назначение каждой строки написанного кода. Если нет понимания, код - в мусорный бак. Его нельзя использовать. Если автор его не понимает, его не поймет никто.

Применительно к политике замещения данных нужно четко понимать что такое политика замещения, какие у нее функции, как эти функции реализуются и т.п.

#4 - 04/20/2013 09:20 AM - Alexander Kamkin

Предлагаю попробовать еще раз. Рассчитываю на серьезный, вдумчивый подход.

#5 - 04/20/2013 09:25 AM - Alexander Kamkin

Также давай договоримся, что работа не считается выполненной, пока не написаны (и не пройдены) тесты, не написаны комментарии.

Символы табуляции нужно заменить на пробелы (1 tab -> 4 spaces).

P.S. Лучше делать медленно, но качественно, чем быстро, но плохо. (в последнем случае не делается ничего.)

#6 - 04/29/2013 06:18 PM - Taya Sergeeva

- Status changed from Open to Resolved

r1250

#7 - 04/30/2013 06:46 AM - Alexander Kamkin

Стало лучше, но вопросы остались

Методы hit и miss не нужны (ни в классе Policy, ни в его наследниках).

```
public abstract void hit(Address address);
public abstract int miss(Address address);
```

choseVictim нужно переименовать в chooseVictim.

```
public abstract void accessLine(int index);
public abstract int choseVictim();
```

При перегрузке методов базового класса нужно указывать аннотацию @Override.

Комментарии к методам accessLine и choseVictim лучше перенести в базовый класс Policy. А в классах-наследниках писать

```
/**
 * {@inheritDoc}
 */
@Override
public void accessLine(int i) { ... }
```

В начало файла нужно поместить комментарий с APL (см. мои файлы).

Не забываем писать автора кода

```
/**
 * This class implements ...
 *
 * @author <a href="mailto:leonsia@ispras.ru">Taya Sergeeva</a>
 */
class X { ... }
```

И, конечно, общий тест на политики вытеснения.

#8 - 04/30/2013 02:55 PM - Taya Sergeeva

r1252

#9 - 06/16/2013 09:46 AM - Alexander Kamkin

- Status changed from Resolved to Closed