

Ассемблер: Домашняя работа 2

ИТМО, КТ

March 7, 2015

1 Задание

Вторая домашняя работа посвящена программированию на языке ассемблера в 64-битном режиме и использованию векторных расширений набора инструкций.

Вам нужно написать библиотеку для работы с матрицами.

Вы можете сами выбрать, как именно хранить структуру, описывающую матрицу (по строкам, по столбцам, с выравниванием, без выравнивания и т. п.). Не забудьте описать это в комментариях к коду.

1.1 Функции

1.1.1 Создание матрицы

```
Matrix matrixNew(unsigned int rows, unsigned int cols);
```

Создает новую матрицу заданного размера и заполняет ее нулями. Память выделяйте с помощью malloc().

1.1.2 Удаление матрицы

```
void matrixDelete(Matrix matrix);
```

Удаляет ранее созданную матрицу. Используйте free().

1.1.3 Функции доступа к структуре

```
unsigned int matrixGetRows(Matrix matrix);
```

```
unsigned int matrixGetCols(Matrix matrix);
```

```
float matrixGet(Matrix matrix, unsigned int row, unsigned int col);  
void matrixSet(Matrix matrix, unsigned int row, unsigned int col,  
float value);
```

1.1.4 Умножение на скаляр

```
Matrix matrixScale(Matrix matrix, float k);
```

Возвращает новую матрицу, содержащую результат умножения матрицы на скаляр. Исходная матрица не изменяется.

1.1.5 Сумма матриц

```
Matrix matrixAdd(Matrix a, Matrix b);
```

Возвращает новую матрицу, содержащую результат сложения двух матриц. Исходные матрицы не изменяются.

Если матрицы имеют разные размеры, то функция должна вернуть 0.

1.1.6 Произведение матриц

```
Matrix matrixMul(Matrix a, Matrix b);
```

Возвращает новую матрицу, содержащую результат умножения матрицы a на матрицу b. Исходные матрицы не изменяются.

Если размеры матриц не позволяют их перемножить, то функция должна вернуть 0.

2 Оформление задания

2.1 Репозиторий

Репозиторий находится по адресу <https://github.com/itmoasm2015/Homework2>.

В папке `include` находится заголовочный файл с прототипам функций, которые вам нужно реализовать.

Свое решение кладите в папку с вашей фамилией внутри папки с номером вашей группы.

2.2 Технические детали

Для операций с вещественными числами используйте векторные расширения системы команд (SSE, SSE2, SSE3, SSSE3, SSE4, AVX). Используйте векторную арифметику! За использование одних только скалярных операций будут снижаться баллы.

Результатом вашей работы должна стать статическая библиотека с именем **libhw.a**, в которой находятся требуемые функции. Библиотека может ссылаться на внешние символы (вам понадобится, например, `malloc` и `free` для выделения памяти). Библиотека должна успешно линковаться в 64-битном режиме.

В репозитории (в папке с вашей фамилией) должен лежать Makefile, при сборке с помощью которого (командой **make**) должна получаться требуемая библиотека (в той же папке).

Статическую библиотеку можно создать командой

```
ar rcs libhw.a object1.o object2.o ...
```

2.3 Сдача задания

После того как вы напишете свой код и закоммитите его в репозиторий, создавайте *issue*, в котором указываете свою фамилию, номер группы и название папки с вашим решением. Указываете в качестве *assignee* меня (мой ник на гитхабе imihajlow).

Как только я проверю ваше задание, я или закрою *issue*, если задание принято, или напишу к нему комментарий по поводу исправлений. Датой сдачи задания считается день создания или последнего изменения вами *issue*, после которого я его закрою.

Напоминаю, что при сдаче до 5 апреля включительно вы получите баллы полностью, а после 5 апреля с коэффициентом 0,6.

2.4 Оценивание

За сданное задание вы получите максимум 15 баллов, из них 7 ставится за комментарии к коду, поэтому обратите особое внимание на читаемость и понятность вашей программы.

Удачи!