Report 4
System and device programming
Prof. Pietro Laface

Nicola Sabino
mat:253839
Date: 04/04/2019

Laboratory 4

`Practice with barriers, memory map and threads.`

- The code is splitted into two parts:
  The firrst one with the regular implemetation using the method `without_th()` and the second one using the method `with_th()`.
- All the global vectors and matrices are initialized by the methods `float *vector_initializer()` and `float **matrix_initializer()`.

  Other two similar methods are provided for debug purposes:
  `float *vector_initializer_debug(int selector)`
  `float **matrix_initializer_debug()`
- K threads are created and they run the same code as follow

```
void *thread_body(){

        /* row selection */
        static int row = -1;
        row++;

        /* perform the product of the i-th row vector of mat and v2 */
        for(int i = 0; i<k ; i++){
                v[row] += mat[row][i] * v2[i];
        }

        pthread_mutex_lock(&dec_counter);
                counter--;
        pthread_mutex_unlock(&dec_counter);

        pthread_mutex_trylock(&final_command);
        if (counter == 0){
       /* last command */
       for(int i=0; i<k; i++){
            sum += v1[i]*v[i];
        }
                printf("the last thread is %lx\n", pthread_self());
    }
        pthread_mutex_unlock(&final_command);

        pthread_exit(NULL);

}
```

- The random value is calculated as follow:

```
float float_rand( float min, float max ){
    float scale = rand() / (float) RAND_MAX; /* [0, 1.0] */
    return min + scale * ( max - min );      /* [min, max] */
}
```

.