

Exercise

Given a text file that contains a list of integers. The first value of the list is the number of entries (numbers) of the rest of the file.

Write a program that reads the file in an array (located in the main memory) and obtains the maximum value using several threads. For obtaining the maximum value the array has to be divided in as many blocks as number of threads. Then, each block will be assigned to a different thread. In this way, each thread computes the maximum value of its assigned array portion. Then, the main program computes the global maximum from the partial values provided by the threads.

Solution

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

struct parameters {
    int start;
    int end;
    int * vec;
    int max;
};
typedef struct parameters param_t;

void * search_max(void * p) {
    int i;
    param_t * param = (param_t*)p;
    param->max = param->vec[param->start];
    for (i=0;i!=param->end;++i) {
        if (param->vec[i]>param->max) {
            param->max = param->vec[i];
        }
    }
    pthread_exit(0);
    return NULL;
}
```



```
int calcula_maximum(int * vec, int tam, int nthreads) {
    param_t * param;
    pthread_t * th;
    int maximum;
    int i;

    param = malloc(sizeof(param_t) * nthreads);
    th = malloc(sizeof(pthread_t) * nthreads);

    for (i=0;i!=nthreads;++i) {
        param[i].start = tam/nthreads;
        param[i].end=(i==nthreads-1)?tam:(tam+1)/nthreads;
        param[i].vec = vec;
        param[i].max=0;
    }

    for (i=0;i!=nthreads;++i) {
        pthread_create(&th[i], NULL, search_max, (void*)&param[i]);
    }

    for (i=0;i!=nthreads;++i) {
        pthread_join(th[i], NULL);
    }

    maximum = param[0].max;
    for (i=1;i!=nthreads;++i) {
        if (param[i].max > maximum)
            maximum = param[i].max;
    }

    free(param);
    free(th);

    return maximum;
}
```

```
int main(int argc, char ** argv) {
    FILE * fich;
    int n, nthreads, i;
    int * vec;
    int ret;
    int maximum;

    if (argc!=3) {
        fprintf(stderr, "Error: incorrect number of arguments \n");
        exit(-1);
    }

    fich = fopen(argv[1], "r");
    if (fich == NULL) {
        perror("It is not posible to open the file");
        exit(-2);
    }

    ret = fscanf(fich, "%d", &n);
    if (ret !=1) {
        fprintf(stderr, "It is not posible to read the file lenght\n");
        exit(-3);
    }

    nthreads = atoi(argv[2]);

    vec = malloc(sizeof(int) * n);
    for (i=0;i!=n;++i) {
        ret = fscanf(fich, "%d", &vec[i]);
        if (ret !=1) {
            fprintf(stderr, "It is not posible to read the element nro %d\n", i);
            fclose(fich);
            free(vec);
        }
    }

    maximum = calcula_maximum(vec,n,nthreads);
    printf("Maximum: %d\n", maximum);

    fclose(fich);
    free(vec);
    return 0;
}
```