#include <iostream>

using namespace std;

//#define DEBUG

//PF stands for pointer functions

typedef void(\*PF)();

//an array of addresses for tasks/jobs. cyclic array.

static const int ntasks = 100;

static PF Task\_Que[ntasks];

//queuer and dispatcher params

static PF \*IN\_PTR = NULL, \*OUT\_PTR = NULL;

//prototypes

void init\_TaskQ(void);

void add2TaskQ(PF task);

void scan\_TaskQ(void);

//tasks

void f1();

void f2();

void f3();

void f4();

void f5();

int main()

{

void myCheck(void);

init\_TaskQ();

add2TaskQ(&f1);

add2TaskQ(&f2);

add2TaskQ(&f3);

add2TaskQ(&f4);

add2TaskQ(&f5);

scan\_TaskQ();

#ifdef DEBUG

myCheck();

#endif

return 0;

}//main

void myCheck(void)

{

for (int i = 0; i < 300; i++)

{

cout << "--------------------------------------" << endl;

cout << "MyCheck iteration : " << i + 2 << endl;

add2TaskQ(&f1);

add2TaskQ(&f2);

add2TaskQ(&f3);

add2TaskQ(&f4);

add2TaskQ(&f5);

scan\_TaskQ();

cout << "--------------------------------------" << endl;

}

}

void init\_TaskQ(void)

{

//init dispatcher and queuer params

OUT\_PTR = IN\_PTR = &Task\_Que[0];

}

void add2TaskQ(PF task)

{

//add a new task to Task\_Que

\*IN\_PTR = task;

//increment ptr

IN\_PTR++;

#ifdef DEBUG

static int ptrIncrementCtr = 0;//debug

ptrIncrementCtr++;//debug

#endif

//check if in array bounds, cyclic

if (IN\_PTR == &Task\_Que[ntasks])

{

IN\_PTR = &Task\_Que[0];

#ifdef DEBUG

ptrIncrementCtr = 0;//debug

#endif

}

//check dispatcher and queuer params, check if queue is full

if (IN\_PTR == OUT\_PTR)

{

exit(1);//exit with 1 for error

}

#ifdef DEBUG

cout << "$ IN\_PTR points to index: " << ptrIncrementCtr << endl;

#endif

}

void scan\_TaskQ(void)

{

while (OUT\_PTR != IN\_PTR)

{

(\*OUT\_PTR)();

OUT\_PTR++;

#ifdef DEBUG

static int ptrIncrementCtr = 0;//debug

ptrIncrementCtr++;//debug

#endif

/\*

please notice that Task\_Que[ntasks] is o.k why? ,

because it is the same as :

ptr = Task\_Que

ptr = ptr+ntasks;

pointer arithmetics...

if i wanted to get something from the array i.e. the value this would fail

but i only need the address...

\*/

if (OUT\_PTR == &Task\_Que[ntasks])

{

OUT\_PTR = &Task\_Que[0];

#ifdef DEBUG

ptrIncrementCtr = 0;//debug

#endif

}

#ifdef DEBUG

cout << "$ OUT\_PTR points to index: " << ptrIncrementCtr << endl;

#endif

}

//empty queue

cout << "# queue is empty #" << endl;

}

void f1()

{

cout << "1" << endl;

}

void f2()

{

cout << "2" << endl;

}

void f3()

{

cout << "3" << endl;

}

void f4()

{

cout << "4" << endl;

}

void f5()

{

cout << "5" << endl;

}

Pls use word + bottom left for the print screens….

