#include <stdio.h>

#include <errno.h>

#include <stddef.h> /\* \_threadid variable \*/

#include <process.h> /\* \_beginthread, \_endthread \*/

#include <time.h> /\* time, \_ctime \*/

#include <semaphore.h>

char listachar[30];

int vetorTeste[3] = {1,1,1};

int indice = 0;

int flag1 = 1;

int flag2 = 0;

int flag3 = 0;

int turn = 0;

int espera = 0;

sem\_t semaforo[4];

sem\_t semaSend[4];

sem\_t semaReceive[4];

int canal[4] = {-1, -1, -1, -1};

int delay\_s(float tempo)

{

time\_t start, end;

long unsigned t;

start = time(NULL);

do{

end = time(NULL);

}while(difftime(end, start)<=tempo);

return 0;

}

void send\_sync(int \*buf, int c){

canal[c] = \*buf;

while(canal[c] != -1);

return;

}

void send\_sync\_s(int \*buf, int c){

canal[c] = \*buf;

sem\_post(&semaSend[c]);

sem\_wait(&semaReceive[c]);

return;

}

void receive(int \*buf, int c){

while (canal[c]==-1);

\*buf = canal[c];

canal[c] = -1;

return;

}

void receive\_s(int \*buf, int c){

sem\_wait(&semaSend[c]);

\*buf = canal[c];

sem\_post(&semaReceive[c]);

return;

}

int teste\_vetor(int vector[]){

int x;

x = vector[0] + vector[1];

return x;

}

int teste\_novovetor(int \*vector1){

int x;

x = \*vector1 + \*(vector1 + 1);

return x;

}

void thread\_code\_a(void \*parg)

{

/\*printf("o valor passado e:%d\n", \*((int\*)parg));

delay\_s(1);

while (indice <= 27){

sem\_wait(&semaforo[0]);

listachar[indice] = 'a';

indice++;

sem\_post(&semaforo[1]);

}

return; \*/

int x = 50;

delay\_s(10);

send\_sync\_s(&x, 0);

// printf("\nTA enviou sua mensagem!\n");

return;

}

void thread\_code\_b(void \*parg)

{

/\* int x[3];

x[0] = \*((int\*)parg);

x[1] = \*((int\*)parg+1);

x[2] = \*((int\*)parg+2);

printf("valor recebido na thread b:%d", x[2]);

while (indice <= 28){

sem\_wait(&semaforo[1]);

listachar[indice] = 'b';

indice++;

sem\_post(&semaforo[2]);

}

return;\*/

int y;

delay\_s(10);

receive\_s(&y, 0);

y = y+10;

delay\_s(5);

send\_sync\_s(&y, 1);

return;

}

void thread\_code\_c(void \*threadno)

{

/\*while (indice <= 29){

sem\_wait(&semaforo[2]);

listachar[indice] = 'c';

indice++;

sem\_post(&semaforo[0]);

}

for (indice = 0; indice <= 29; indice++)

printf("%c\n", listachar[indice]);

espera = 1; \*/

int z;

receive\_s(&z, 1);

printf("\nvalor recebido de b e:%d\n", z);

return;

}

void start\_thread()

{

long int thread\_id;

int param = 125;

int vetorTesteTH[3] = {4,5,6};

int \* pparam = &param;

int \* vparam = vetorTesteTH;

#if defined(\_\_WIN32\_\_)

if ((thread\_id = \_beginthread(thread\_code\_a,4096, (void \*) pparam)) == (unsigned long)-1)

#else

if ((thread\_id = \_beginthread(thread\_code\_a,4096, (void \*) pparam)) == -1)

#endif

{

printf("Unable to create thread a.\n");

return;

}

printf("Created thread a.\n");

#if defined(\_\_WIN32\_\_)

if ((thread\_id = \_beginthread(thread\_code\_b,4096, (void \*) vparam)) == (unsigned long)-1)

#else

if ((thread\_id = \_beginthread(thread\_code\_b,4096, (void \*) vparam)) == -1)

#endif

{

printf("Unable to create thread b.\n");

return;

}

printf("Created thread b.\n");

#if defined(\_\_WIN32\_\_)

if ((thread\_id = \_beginthread(thread\_code\_c,4096, NULL)) == (unsigned long)-1)

#else

if ((thread\_id = \_beginthread(thread\_code\_c,4096, NULL)) == -1)

#endif

{

printf("Unable to create thread c.\n");

return;

}

printf("Created thread c.\n");

}

int main(void)

{

int i,k,l;

long int thread\_id;

k = teste\_vetor(vetorTeste);

for(int i = 0; i<=3; i++){

sem\_init(&semaforo[i], 0, 0);

sem\_init(&semaSend[i], 0, 0);

sem\_init(&semaReceive[i], 0, 0);

}

// printf("valor da soma do vetor1 e:%d\n", k);

l = teste\_novovetor(vetorTeste);

// printf("valor da soma do vetor2 e:%d\n", l);

sem\_post(&semaforo[0]);

start\_thread();

system("PAUSE");

return 0;

}