// DeadlockAvoidance.h : Include file for standard system include files,

// or project specific include files.

//Fredrick Onduso Ondieki

// CST-221 Deadlock avoidance

// John Zupan

// March 24 2019

//Implementing a deadlock avoidance; Use of Timers

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <unistd.h>

int availableResource = 4;

int reqResource;

int aNeed = 5;

int bNeed = 10;

int aHave = 3;

int bHave = 3;

int main() {

// Show process A resources and total resources available

printf("Process A has %d resources and needs a total of %d resources.\n", aHave, aNeed);

printf("Process A will now request more resources.\n");

printf("\n");

// Process A requesting resources loop until all resources are obtained

while (aNeed > aHave) {

// If this process was to run again it would have the if statement like process B

aHave++;

availableResource--;

reqResource++;

// Display resource info

printf("Process A has %d resources and needs a total of %d resources.\n", aHave, aNeed);

printf("There are %d total resources available left to the system.\n", availableResource);

printf("\n");

}

// Process A will now run

printf("Process A has all necessary resources to run.\n");

printf("\n");

// Process B requesting resources loop until all resources are obtained

while (bNeed > bHave) {

// Checks to see if there are enough resources, if not tries again in 2 seconds

if (availableResource <= 0) {

printf("not enough resources, try again\n");

int s = 0;

// Timer

while (s <= 2) {

printf("\r"); // move cursor to position 0

printf("%d Seconds\n", s);

sleep(1); // Time in seconds

s++;

}

// The Process A will finish and free resources making them available for B

printf("\n");

reqResource = 0;

aHave = 0;

availableResource = 5;

printf("Process A is complete and has released its resources\n");

printf("\n");

// Process B aquiring the necessary resources from Process A

}

else {

bHave++;

availableResource--;

reqResource++;

// Displays resource aquisition progress

printf("Process B has %d resources and needs a total of %d resources.\n", bHave, bNeed);

printf("There are %d total resources available left to the system.\n", availableResource);

printf("\n");

}

}

// Process B runs after having resources

printf("Process B has all necessary resources to run.\n");

printf("\n");