PCO003

Lab 2 Criando seu primeiro kernel

Este lab será realizado inteiramente no CodeBlocks.

1)Crie um novo projeto com 2 arquivos kernel.c e kernel.h

No kernel.h:

#ifndef KERNEL\_H

#define KERNEL\_H

//return codes

#define SUCCESS 0

#define FAIL 1

#define REPEAT 2

//function pointer declaration

**typedef** char**(\***ptrFunc**)(**void**);**

//process struct

**typedef** struct **{**

ptrFunc func**;**

**}** process**;**

//Kernel Functions

char kernelInit**(**void**);**

char kernelAddProc**(**process **\*** newProc**);**

void kernelLoop**(**void**);**

#endif

Adicione no kernel.c:

#include "kernel.h"

#define POOL\_SIZE 10

process **\*** pool**[**POOL\_SIZE**];**

int start, end;

char kernelInit**(**void**){**

start **=** 0**;**

end **=** 0**;**

**return** SUCCESS**;**

**}**

char kernelAddProc**(**process **\*** newProc**){**

//checking for free space

**if** **(** **((**end**+**1**)%**POOL\_SIZE**)** **!=** start**){**

pool**[**end**]** **=** newProc**;**

end **=** **(**end**+**1**)%**POOL\_SIZE**;**

**return** SUCCESS**;**

**}**

**return** FAIL**;**

**}**

void kernelLoop**(**void**){**

**for(;;){**

//Do we have any process to execute?

**if** **(**start **!=** end**){**

**--------------------------**

//check if there is need to reschedule

**if** **(**pool**[**start**]->**func**()** **==** REPEAT**){**

kernelAddProc**(**pool**[**start**]);**

**}**

//prepare to get the next process;

start **=** **(**start**+**1**)%**POOL\_SIZE**;**

**}**

**}**

**}**

2) Crie o arquivo main.c que deverá: adicionar quatro processos.

a) O primeiro imprime seu nome e retornar SUCCESS.

b) O segundo imprime seu número de matricula e retornar SUCCESS.

c) O terceiro deve imprimir “UNIFEI” retornar REPEAT.

d) O quarto deve imprimir “PCO003” e quantas vezes já foi executado, retornando REPEAT no fim.

3) Modifique o programa anterior para incluir um novo membro na estrutura do processo. Ele será um **int** com o nome **priority**. A função kernelLoop() deverá procurar quem é o processo com maior prioridade, trocá-lo de posição com o primeiro e executá-lo. Se ele quiser ser reexecutado ele deverá ser adicionado novamente no fim da fila.