**Rafael Giordano Matesco 50210686**

**1HID**

**Parte1:**

**https://youtu.be/ulRckR53tNA**

**Parte2:** **https://www.youtube.com/watch?v=kqAZ47FPAl0**

**PARTE 1**

//PARTE 1

#include <stdio.h>

#include <stdlib.h>

int opcao;

int array[10];

int z;

int vezesproc = 9;

int x = 0;

int aa;

void menu(){

fflush(stdin);

opcao = 0;

printf("========= MENU =========");

printf("\n\nObservacoes:\n-Nao use valores com mais de 10 algarismos, pois assim o programa corre risco de nao funcionar direito\n-Maximo de 10 processos, alem desse valor o sistema nao eh capaz de processar\n");

printf("\n-Inserir novo processo (1)");

printf("\n-Eliminar um processo (2)");

printf("\n-Ordenar processos (3) ");

printf("\n-finalizar programa (4)\n");

do{

scanf("%d", &opcao);

if((opcao>4) || (opcao<1)){

printf("digite um numero valido (1,2, 3 ou 4)\n");

}

}while((opcao>4) || (opcao<1));

printf("\n");

}

void novoprocesso(){

fflush(stdin);

if(vezesproc <= 0){

vezesproc =9;

}

int resp;

printf("\nDigite o tamanho do processo:\n");

scanf("%d", &array[x]);

printf("\n");

x++;

}

void processar(){

fflush(stdin);

//eliminando o menor numero

int menor = 1000000000;

int cont;

int aux;

for(cont = 0; cont <= vezesproc; cont++){ //encontrando menor numero

if ((array[cont] < menor) && (array[cont] > 0)){

menor = array[cont];

}

}

for(cont = 0; cont <= vezesproc; cont++){ //colocando o menor numero na ultima posicao (eliminando)

if (array[cont] == menor){

array[cont] = -1; //eliminacao

aux = array[vezesproc];

array[vezesproc] = array[cont];

array[cont] = aux;

}

}

if((vezesproc >= 0) && (array[vezesproc] != 0)){

printf("array processado:\n");

for(aa=0; aa<=9; aa++){

if(array[aa] > 0){

printf("|");

printf("%d| ", array[aa]);

}

}

}else{

printf("seu array de processos esta vazio!");

}

printf("\n");

vezesproc = vezesproc -1;

}

void reordenar(){

fflush(stdin);

int conta = 10;

int y;

int aux2;

while(conta > 1){

for(y=0; y<conta-1; y++){

if(array[y] > array[y+1]){

aux2 = array[y];

array[y] = array[y+1];

array[y+1] = aux2;

}

}

conta = conta-1;

}

if((vezesproc >= 0) && (array[vezesproc] != 0)){

printf("array ordenado!\n");

for(aa=0; aa<=9; aa++){

if(array[aa] > 0){

printf("|");

printf("%d | ", array[aa]);

}

}

}else{

printf("seu array de processos esta vazio!");

}

printf("\n");

}

//Main---------------------------------------------------------------------------------------

main(){

do{

system("cls");

menu();

system("cls");

if(opcao == 1){

novoprocesso();

}else{

if(opcao == 2){

processar();

system("pause");

}else{

if(opcao == 3){

reordenar();

system("pause");

}else{

if(opcao == 4){

system("pause");

return(0);

}

}

}

}

z = 1;

}while(z = 1);

}

**Parte 2**

//PARTE 2

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char progs[11];

char progsAptos[11];

char progsSusp[11];

char progsTerminados[513];

int opcao;

int contNovo;

int contApto;

int contSusp;

char deseja[11];

int contd;

//menu

void Menu(){

opcao = 0;

puts("==========MENU==========\n");

puts("OBS: o processamento maximo eh 10 programas\nOBS: nao usa a letra (z) minuscula\n");

printf("1-Abrir novo programa \n2-Listar programas na memoria \n3-Listar programas de aptos \n4-Listar programas na fila de suspensos \n");

puts("5-Enviar programa da memoria para fila de aptos \n6-Executar programa \n7-Tirar programa da fila de suspensos");

scanf("%d", &opcao);

}

//1

void novoProg(){

if(contNovo<10){

int x;

puts("Adicione um programa na fila: ");

progs[contNovo] = getchar();

fflush(stdin);

for(x=0; x<contNovo; x++){

if((progs[contNovo] == progs[x]) || (progs[x] == 'z')){

puts("Esse programa ja foi aberto ou eh um nome invalido, tente novamente");

system("pause");

fflush(stdin);

contNovo--;

}

}

}else{

puts("Sua memoria esta cheia!");

system("pause");

}

fflush(stdin);

contNovo++;

}

//2

void listarProg(){

int x;

printf("|");

for(x=0; x<contNovo; x++){

if(progs[x] != 'z'){

printf(" %c |", progs[x]);

fflush(stdin);

}

}

puts("");

system("pause");

fflush(stdin);

}

//3

void listarProgApto(){

fflush(stdin);

}

//4

void listarProgSusp(){

fflush(stdin);

}

//5

void enviarProgApto(){

int x, y, z, random, erro;

if((contSusp + contApto) < contNovo){

printf("|");

for(x=0; x<contNovo; x++){

if(progs[x] != 'z'){

printf(" %c |", progs[x]);

fflush(stdin);

}

}

puts("\nQual destes programas voce deseja executar?");

deseja[contd] = getchar();

system("cls");

fflush(stdin);

for(x=0; x<contNovo; x++){

if(deseja[contd] == progs[x]){

fflush(stdin);

srand (time(NULL));

random = rand() % 2; //SORTEANDO PARA SER SUSPENSO OU APTO

fflush(stdin);

if(random % 2 == 0){

fflush(stdin);

progsAptos[contApto] = progs[x];//PROG NA MEMORIA VIRA APTO

progs[x] = 'z';

contApto++;

puts("programa adicionada na fila de Aptos\n");

fflush(stdin);

}else{

if(random % 2 != 0){

fflush(stdin);

progsSusp[contSusp] = progs[x];//PROG NA MEMORIA VIRA SUSPENSO

progs[x] = 'z';

contSusp++;

puts("programa suspendido\n");

fflush(stdin);

}

}

}

}

fflush(stdin);

if((deseja[contd] != progs[0]) && (deseja[contd] !=progs[1]) && (deseja[contd] !=progs[2]) && (deseja[contd] !=progs[3]) && (deseja[contd] !=progs[4]) && (deseja[contd] !=progs[5]) && (deseja[contd] !=progs[6]) && (deseja[contd] !=progs[7]) && (deseja[contd] !=progs[8]) && (deseja[contd] !=progs[9])){

printf("Digite o nome de um programa que exista\n");

}

}else{

puts("Voce ja enviou todos os programas!");

}

contd++;

system("pause");

fflush(stdin);

}

//6

void executarProg(){

fflush(stdin);

}

//7

void tirarProgSusp(){

fflush(stdin);

}

main(){

int infinito = 1;

while(infinito > 0){

system("cls");

fflush(stdin);

Menu();

if(opcao == 1){

system("cls");

fflush(stdin);

novoProg();

}else{

if(opcao == 2){

system("cls");

fflush(stdin);

listarProg();

}else{

if(opcao == 3){

system("cls");

fflush(stdin);

listarProgApto();

}else{

if(opcao == 4){

system("cls");

fflush(stdin);

listarProgSusp();

}else{

if(opcao == 5){

system("cls");

fflush(stdin);

enviarProgApto();

}else{

if(opcao == 6){

system("cls");

fflush(stdin);

executarProg();

}else{

if(opcao == 7){

system("cls");

fflush(stdin);

tirarProgSusp();

}

}

}

}

}

}

}

}

}