

Atividade - Revisão de Conceitos de Programação Concorrente e Paralela

Grupo:

- Caroline Yumi Uehara,
- José Augusto Barros Minhoto,
- Leonardo de Jesus Diz Conde,
- Marcelo Maués Botelho de Souza

Atividade 1 – Conta corrente

- 1 - Sim, ocorre um erro inesperado de saldo insuficiente
- 2 - Erro de sincronização de threads que tentaram alterar um valor ao mesmo tempo.
- 3 - Sim, devemos alterar o main para a seguinte forma:

```
while (!fim) {  
    iniciarSaldo(&conta1); // inicia saldo com 5000  
  
    // threads que realizam as operações na mesma conta corrente  
  
    pthread_create(&thread1, NULL, depositar,&conta1); // operação de deposito (R$300,00)  
    pthread_join(thread1,NULL); //esperar a thread1 acabar para criar a segunda  
  
    pthread_create(&thread2, NULL, retirar,&conta1); // operação de retirada (R$500,00)  
    pthread_join(thread2,NULL);  
  
    tentativas++;  
  
    if (conta1.saldo != 4800) fim = TRUE;  
    else mostrarCC(conta1);  
}
```

Atividade 2 – Calcular Média

- 1 - Não, o programa calcula a média aritmética de cada arquivo
- 2 - Calcular as notas de maneira paralela (com thread) ao invés de sequencial como o programa está fazendo.
- 3 – Calculando a média com todos os resultados das médias de cada arquivo calculado

O programa a seguir foi criado para responder as questões 2 e 3 dessa atividade:

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>


#define TIME 10


void calcularMedia(FILE *arq,int pfd[], int i);


int main(int argc, char * argv[]) {

    float resultado = 0.0,media;

    int i,k,pid,pfd[2];

    FILE *file;

    if (argc > 1){

        if (pipe(pfd) < 0) exit(1);

        printf("\nProcurado e lendo os arquivos:\n\n");

        for (i=1; i<argc; i++) {

            file = fopen(argv[i], "r");

            if(!file){

                printf("ERROR: Não foi possível ter acesso ao arquivo: '%s' (num = %d)\n",argv[i],i);

            }else{

                pid =fork();

                if (pid==0){

                    printf("O arquivo '%s' (num = %d) foi acessado com sucesso!!!\n",argv[i],i);

                    calcularMedia(file,pfd,i);

                    exit(0);

                }

            }

        }

        for(k=0;k<(argc-1);k++){

            wait(0);

            close(pfd[1]);

            read(pfd[0],&media,sizeof(media));

            resultado = resultado + media;

        }

    }
```

```

printf("\nMedia geral:\n\nA media das medias equivale a: %3.2f;\n\n",resultado/(argc-1));

}else{

    printf("Arquivos de entrada nao localizados!!\n");

    exit(-1);

}

exit(0);

}

void calcularMedia(FILE *arq,int pfd[], int num){

    float media = 0.0,resultado = 0.0,nota;

    char buf[1000];

    int i=0;

    while (fgets(buf,1000, arq)!=NULL){

        nota = atof(buf);

        media = media + nota;

        i++;

    }

    sleep(TIME);

    close(pfd[0]);

    sleep(TIME);

    resultado =media/i;

    if(num==1) printf("\nLeituras realizadas:\n\n");

    printf("A media do arquivo %d equivale a: %3.2f;\n",num,resultado);

    write(pfd[1],&resultado,sizeof(resultado));

    wait(NULL);//espera escrever

}

```