

# Lista de Exercício - Threads

---

Author: Denis Moura

Data: 15/10/2019

---

## Resolução

### 1. Como uma aplicação pode implementar concorrência em um ambiente monothread

A concorrência permite que varias tarefas sejam executadas simultaneamente e termine em espaços de tempo semelhantes. Em ambientes single core, as tarefas não são executadas no mesmo exato momento, porém mais de uma tarefa é processada em um mesmo intervalo de tempo, sem a necessidade de esperar que uma tarefa seja finalizada para dar inicio a outra.

### 2. Explique a diferença entre unidade de alocação de recursos e unidade de escalonamento

Em ambientes monothread, o processo é ao mesmo tempo uma unidade de alocação de recursos e escalonamento. Em um ambiente multithread a unidade de alocação de recursos é o processo, onde todos os seus threads compartilham o espaço de endereçamento, descritores de arquivos e dispositivos de E/S. Cada thread respresenta uma unidade de escalonamento e, nesse caso, o sistema não seleciona um processo para a execução, mas sim um de seus threads.

### 3. Desenvolva um programa multithread que verifica 4 sensores em um tonel de óleo. Cada sensor deve ser utilizada uma thread

```
/**
 * Name:          verifySensor
 * Description:    Monitora 4 sensores utilizando multithreads
 * Author:        Denis Moura
 * Data:          13-10-2019
 */

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

/** Define o numero de threads disponíveis pra uso*/
#define NUM_THREADS 3

/** Função que executa as Threads*/
void *monitorSensor(void *args);

int main(void){

    pthread_t threads [NUM_THREADS];
    int i, args [NUM_THREADS];

    for(i = 0; i < NUM_THREADS; i++){
```

```
        args[i] = i++;
        pthread_create(&threads[i], NULL, monitorSensor, (void *)&args[i]);
    }

    for(i = 0; i < NUM_THREADS ;i++){

        pthread_join(threads[i],NULL);

    }

    exit(NULL);
}

void *monitorSensor(void * args){

    int pvalor = (int)(*args);

    printf("Iniciando a thread %i.Monitorando sensor %i...\n", *pvalor,*pvalor+1
);

    switch (pvalor){
        case 0:
            printf("Verificando sensor 1");
            break;
        case 1:
            printf("Verificando sensor 2");
            break;
        case 2:
            printf("Verificando sensor 3");
            break;
        case 3:
            printf("Verificando sensor 4");
            break;
        default:
            break;
    }

}
```

#### 4. Dê exemplos do uso de threads no desenvolvimento de aplicativos, como editores de textos e planilhas eletrônicas

O editor de notas nativo dos sistemas operacionais é um bom exemplo do uso de threads. Esses programas são considerados multitarefas pois são sistemas em que mais de um programa pode estar em execução simultaneamente, utilizando o tempo de processamento ocioso para trocar de contexto e executar outras tarefas. Dessa forma, você pode usar um mesmo aplicativo para editar varios arquivos diferentes ao mesmo tempo, impedindo que o processo fique subutilizado e ocioso na maior parte do tempo esperando outros componetes, como dispositivos de I/O.