

- 1 – Crie um programa com duas threads. As medidas de um terreno retangular devem ser lidas. Um thread deve calcular a área do terreno e outro o perímetro. Todos os resultados obtidos devem ser mostrados ao usuário.
- 2 – Escreva um programa formado por 3 threads, que executam um laço de repetição de N interações. Neste laço, cada thread imprime sua identificação. A partir da execução do programa, identifique como acontece o escalonamento dos threads.
- 3 – Escreva um programa formado por várias threads, que executam um laço de repetição de N interações para incrementar em 1 uma variável compartilhada. Ao término da execução, verifique o valor final da variável compartilhada.
- 4 – Escreva um programa com duas threads que compartilham um vetor de inteiros tamanho 5 e uma outra variável inteira que conta quantos elementos há no vetor. Uma das threads escreve um dado no vetor e a outra remove um dado no vetor. A execução das threads é contínua.
  - a) Verifique a execução do programa
  - b) Os valores apresentados nos dados compartilhados são corretos
  - c) Neste problema, as threads precisam ser sincronizadas?
- 5 – Considere o problema de somar dois vetores de tamanho N. Desenvolva um algoritmo que permita a criação de diferentes quantidades de threads para efetuar a soma de dois vetores e mostrar o vetor resultante.
  - d) Efetue a medida do tempo de execução deste programa, usando a função *gettimeofday*, conforme o quadro 1.
  - e) Compare a execução deste algoritmo com threads, com a execução do algoritmo da lista 2 (com processos). Qual versão mostrou menor tempo de execução?
  - f) Neste problema, existiu acesso concorrente aos dados pelos diferentes fluxos de execução?

```
#include <sys/time.h>
...

struct timeval tpI, tpF;
int sec, usec
...

gettimeofday(&tpI, NULL);

//processamento

gettimeofday(&tpF, NULL);

sec = tpF.tv_sec - tpI.tv_sec;
usec = tpF.tv_usec - tpI.tv_usec;
if (usec<0){
    usec += 1000000L;
    sec -= 1;
}
printf("tempo: %d s %d us\n", sec, usec);
```

Quadro 1 – Exemplo de uso da função *gettimeofday*