ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES II

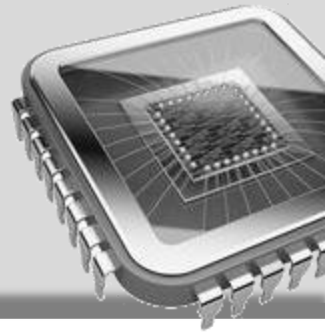
AULA 08: PROGRAMANDO COM THREADS EM LINGUAGEM C (Continuação)

Prof. Max Santana Rolemberg Farias
max.santana@univasf.edu.br
Colegiado de Engenharia de Computação

GESE
Grupo de Engenharia de Sistemas Embarcados

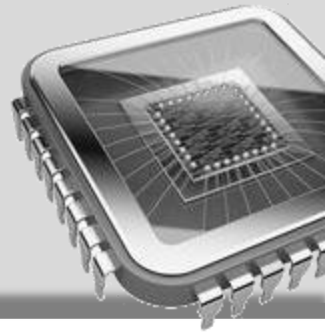
CEComp
Engenharia da Computação
UNIVASF

PROGRAMPROGRAMANDO COM THREADS PTHREADS API: SEÇÃO CRÍTICA

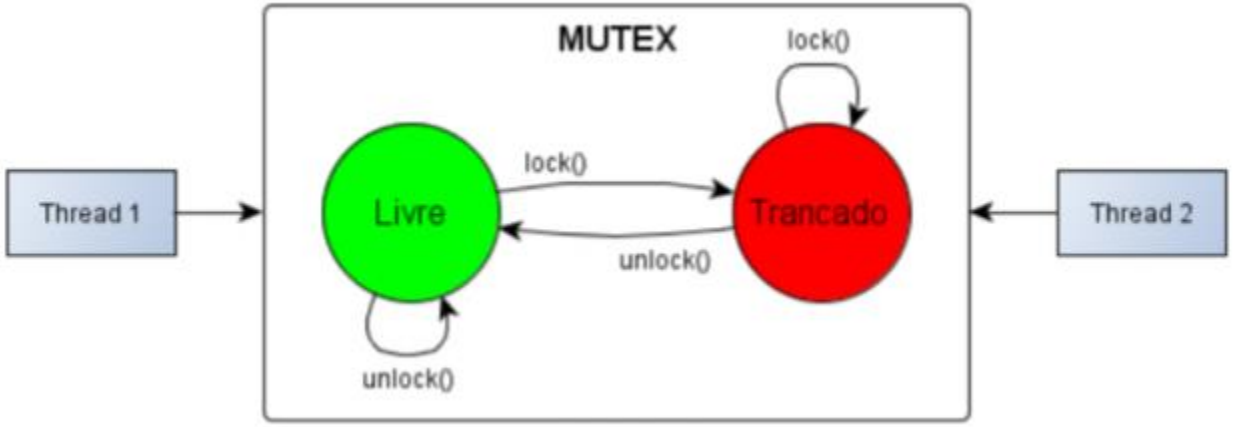


- Segmento de código que só pode ser executado por uma thread
- Em pthreads a seções críticas são implementadas usando **semáforos mutex**
  - Semáforos mutex possuem dois estados apenas: livre (locked) e bloqueado (unlocked)
  - A qualquer momento, somente uma thread pode bloquear (obter o mutex)
- Thread requisita semáforo no inicio da seção, ficando bloqueado até conseguir obtê-lo
- Libera o mutex no final da seção

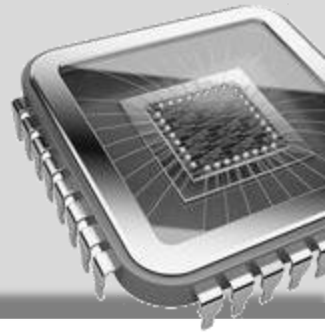
PROGRAMPROGRAMANDO COM THREADS PTHREADS API: EXCLUSÃO MÚTUA



- Exclusão mútua (mutex) é uma técnica usada em programação concorrente para evitar que duas threads tenham acesso simultaneamente a uma região crítica.

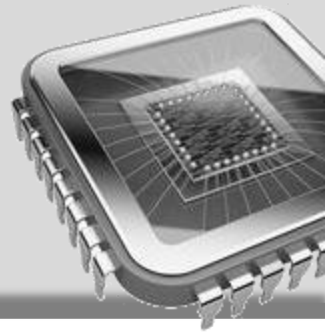


PROGRAMPROGRAMANDO COM THREADS PTHREADS API: MUTEX



- Criar variável mutex
  - int pthread_mutex_init(pthread_mutex_t *mutex_lock, const pthread_mutexattr_t *lock_attr)
- Bloquear uma variável mutex
  - int pthread_mutex_lock(pthread_mutexx_t *mutex_lock)
- Liberar uma variável mutex
  - int pthread_mutex_unlock (pthread_mutex_t *mutex_lock)

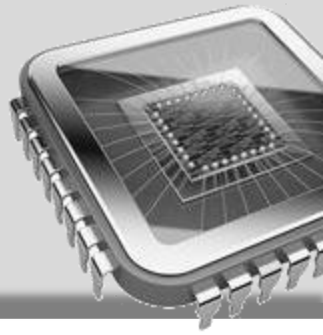
PROGRAMPROGRAMANDO COM THREADS PTHREADS API: EXERCÍCIO 1



a) Implemente um programa que informe quantas vezes o elemento 5 acontece em um vetor com 9000000 posições.
   - Compilei com o gcc, sem optimizações e com a opção de gerar informação para o gprof
     - $ gcc -pg exercicio1a.c –o exercicio1a
     - $ ./exercicio1a
     - $ gprof

b) Implemente um programa paralelo com 5 thread que informe quantas vezes o elemento 5 acontece em um vetor com 9000000 posições.
   - Compilei com o gcc, sem optimizações e com a opção de gerar informação para o gprof
     - $ gcc -pg exercicio1b.c –o exercicio1b -lpthread
     - $ ./exercicio1b
     - $ gprof

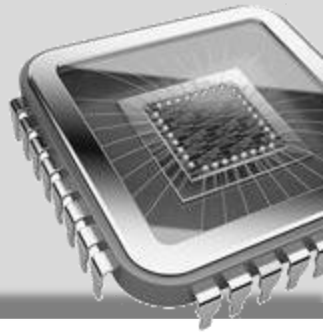
PROGRAMPROGRAMANDO COM THREADS PTHREADS API: EXERCÍCIO 2



a) Implemente um programa que multiplique duas matrizes 1000x1000. (sequencial)

b) Implemente um programa paralelo que multiplica duas matrizes 1000x1000.

PROGRAMPROGRAMANDO COM THREADS
DESEMPENHO DE ALGORITMOS PARALELOS

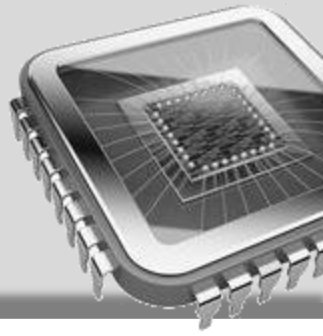


- É importante entender até que ponto é vantajoso utilizar programas paralelos.

- O objetivo é determinar os benefícios do paralelismo aplicados a um problema considerado.

- E também o quanto o algoritmo é capaz de continuar eficiente.

PROGRAMPROGRAMANDO COM THREADS SPEEDUP E EFICIÊNCIA



- O speedup (S) diz quantas vezes o programa paralelo é mais rápido que o programa serial.

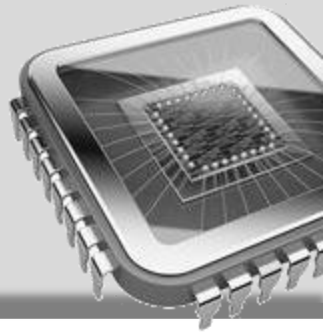
$$S = Ts/Tp$$

- A eficiência (E) é uma medida normalizada de speedup que indica o quão efetivamente cada processador é utilizado.

$$E = S/P = (Ts/Tp)/P = Ts/(P.Tp)$$

  - P é o número de threads
  - Um speeduo com valor igual a P, tem uma eficiência igual a 1, ou seja, todos os processadores são utilizados e a eficiência é linear.

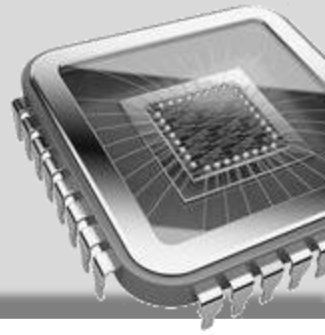
PROGRAMPROGRAMANDO COM THREADS
SPEEDUP E EFICIÊNCIA: EXERCÍCIO



a) Calcule o speedup e a Eficiência do exercício 1.

b) Calcule o speedup e a eficieência do exercício 2.

PROGRAMPROGRAMANDO COM THREADS
LEI DE AMDAHL



- Segundo a Lei de Amdahl, a velocidade de processamento paralelo é limitada a porção sequencial do programa.

- Essa porção do programa que não pode ser paralelizada limitará o aumento de velocidade disponível com o paralelismo.

PROGRAMPROGRAMANDO COM THREADS
LEI DE AMDAHL