

Introdução a Programação Concorrente

Aula – 02

Prof. André Luis Meneses Silva

Agenda

- Pthreads
- Trabalhando com Threads
- Exclusão Mútua
- Semáforos

Pthreads

- Historicamente, fabricantes de hardware implementam suas próprias versões de threads.
 - Implementações diferentes não são compatíveis.
- Pthreads (Posix Threads)
 - Padrão especificado em 1995 para prover portabilidade no uso de sistemas unix
 - Definidos como um conjunto de bibliotecas C (pthread.h).

Criando uma Pthread

- Antes, avaliar se a biblioteca está disponível seu S.O.
- pthread_intro
- Não esquecer de utilizar a diretiva lpthread
 - gcc "-lpthread"

Criando uma Pthread

- Passos
 - pthread_t (tipo pthread_t).
 - Definir a função que a thread vai executar
 - void* funcao (void*)
 - Chamar função
 - pthread_create (thread, attr, routine, arg)
 - Caso queira aguardar a execução das threads criadas, chamar a função
 - pthread_join (threadid, status);
- pthread_intro

Cancelando uma Pthread

- Podemos cancelar a execução de uma thread.
- Para isso usamos as seguintes funções:
 - `pthread_cancel(t);`
 - Cancela a thread de nome t
 - `pthread_setcancelstate(PTHREAD_CANCEL_ENABLE, NULL);`
 - Habilita o suporte para cancelamento de threads.
 - `pthread_setcanceltype();`
 - Diz qual tipo de cancelamento ela dá suporte
- `cancela_thread.c`

Exclusão Mútua

- Seção crítica
 - É uma seqüência ou bloco de código em um Thread que
 - Referencia uma ou mais variáveis (leitura, atualização ou escrita) enquanto uma delas pode ser alterada por outra Thread
 - Altera uma ou mais variáveis que possivelmente são referenciadas (leitura, atualização ou escrita) por outra Thread

Exclusão Mútua

- Usa um Estrutura de Dados enquanto alguma parte dela está sendo alterada por um outra Thread
- Altera qualquer parte de uma Estrutura de Dados enquanto ela esteja sendo usada por outra Thread

Exclusão Mútua

- Exclusão Mútua
 - Para prevenir condições de corrida e atualizações perdidas, deve haver exclusão mútua
 - Consiste em estabelecer protocolos de entrada e saída para evitar que mais de uma thread entre em sua seção crítica simultaneamente

Exclusão Mútua

- Protocolos de Entrada e Saída

```
// Para todas as threads
while (true) {

// código fora da seção crítica

    entrarSC(i); // Protocolo de entrada

// código dentro da seção crítica

    sairSC(i);   // Protocolo de saída
}
```

Exclusão Mútua

- Para fazer exclusão mútua em C
 - `pthread_mutex_t` (tipo mutex)
 - `pthread_mutex_init()` (inicialização)
 - `pthread_mutex_lock()` (aloca mutex)
 - `pthread_mutex_unlock()` (desaloca mutex)
- `exemplo_mutex`

Semáforos

- São importantes para programação concorrente e sistemas operacionais
 - Protegem e controlam o acesso a seções críticas
 - Provêem um mecanismo para sincronização de threads

Semáforos

- Foi projetado em nos anos 60 por Dijkstra como uma ferramenta para resolver problemas de sincronização em programas com múltiplas threads
- Consiste em um TAD com apenas 2 operações:
 - P (*passeren*, passar em alemão)
 - V (*vrygeven*, liberar em alemão)
- Tanenbaum chama P de Down e V de Up

Semáforos

- Informalmente, é semelhante a uma variável inteira especial que é acessada apenas através de P e V
- Quando P é chamado e o valor do semáforo é positivo, o valor é decrementado (de forma atômica). Caso contrário, a thread espera.
- Quando V é chamado e não há thread esperando, o valor do semáforo é incrementado. Caso contrário, uma das threads esperando é liberada (não é FIFO)

Semáforos

- Implementação de P(S)
 1. Desabilitar interrupções (garantir que o acesso é atômico)
 2. SE $S > 0$ ENTÃO $S = S - 1$
 3. SENÃO
 - Colocá-la na lista sobre S
 - Mudar o estado da thread para esperando
 - Escalonar outra thread
- Habilitar interrupções

Semáforos

- Implementação de $V(S)$
 1. Desabilitar interrupções (garantir que o acesso é atômico)
 2. SE $(S == 0)$ E (lista sobre S não está vazia)
ENTÃO
 - Selecionar uma thread da lista
 - Mudar o estado da thread para pronta
 - SENÃO $S = S + 1$
 - Habilitar interrupções

Semáforos

- Tipos de Semáforos
 - Binários
 - São semáforos que só podem assumir os valores 0 e 1
 - Contador
 - É um tipo de semáforo que pode assumir valores de 0 a n.

Semáforos Binário

- Utilização

Semáforo Binário S

Uma Thread executa:

if (!condição) P(S);

Semáforo Binário S

Outra Thread executa:

V(S)

Semáforos

- Podemos utilizar semáforos através da biblioteca semaphore.h
 - sem_t (tipo semáforo)
 - int sem_init(sem_t *sem, int pshared, unsigned int value);
 - (inicializa o semáforo);
 - sem_post
 - Desaloca o semáforo;
 - int sem_wait(sem_t *sem);
 - Aloca o semáforo;
 - sem_destroy
- exemplo_semaforo

Multiplicação de matrizes

- exemplo_matrix.

Referências

- Notas de aula, programação concorrente.
 - Prof. Alberto Costa Neto
- Posix Threads Programming
 - <https://computing.llnl.gov/tutorials/pthreads/>