

Redes 1

Atividade 5

César Kallas

RA: 02099224

cesarkallas at gmx.net

<http://cesarkallas.soulivre.org>

1. Explique sucintamente para que serve cada função pthread_* usada no servidor TCPmtechod.

pthread_attr_init(&ta);

Inicializa o atributo do tipo thread (ta) com um valor default (PTHREAD_CREATE_JOINABLE).

pthread_attr_setdetachstate(&ta, PTHREAD_CREATE_DETACHED);

Seta o atributo do tipo thread (ta) para outro valor (PTHREAD_CREATE_DETACHED).

Nesse estado setado, os recursos da thread são imediatamente liberados quando ela termina, mas a função pthread_join não pode ser usada para sincronizar o termino de uma thread.

pthread_mutex_init(&stats.st_mutex, 0);

Mutex é um dispositivo do sistema operacional que controla o compartilhamento de estrutura de dados que são acessados por várias linhas de execução (threads ou processos), bloqueando a alteração dessa estrutura por várias threads ao mesmo tempo, perdendo a integridade dos dados. Esa função inicializa um novo mutex do tipo pthread_mutex_t que está presente na estrutura de dados stats e por uma constante que indentifica o atributo do objeto do tipo mutex.

pthread_mutex_lock(&stats.st_mutex);

Bloqueia via mutex o acesso a estrutura de dados stats, assim, somente o thread que fez o bloqueio consegue alterar dados nessa estrutura.

pthread_mutex_unlock(&stats.st_mutex);

Desbloqueia via mutex o acesso a estrutura de dados stats, assim, qualquer thread pode alterar dados dessa estrutura.

pthread_create(&th, &ta, (void * (*)(void *))prstats, 0) < 0)

Cria uma nova thread (linha de execução). O atual processo passa a ter mais de uma linha de execução. O primeiro parametro é do tipo (pthread_t), o segundo parametro são os atributos dessa thread a ser criada, o terceiro parametro é a função que a thread passa a executar (prstats) e o último parametro é usado para o inicio da thread.

2. Algumas funções de biblioteca não são thread safe. O que isso significa? Dê um exemplo de função que se enquadra nessa categoria. É possível a várias linhas de execução (threads) de um mesmo processo usar esse tipo de funções? Como?

Thread-safe descreve uma parte ou rotina de um programa que pode ser chamada por multiplas linhas de um programa com garantia de interação entre elas. Significa que uma thread não interferir em dados de uma outra. Ou seja, refere-se a uma aplicação que tem a habilidade de executar multiplas threads simultaneamente sem danificar / atrapalhar um dado compartilhado entre elas ou criar uma condição de corrida.

Por exemplo, suponha que uma biblioteca ou rotina acesse ou modifique uma estrutura global ou um lugar qualquer na memória. Se duas threads ao mesmo tempo chamarem essa rotina é possível que elas tentem modificar essa estrutura global ou parte da memória ao mesmo tempo. Se essa rotina não tiver em si uma parte de sincronização para prevenir a corrupção desses dados, então ela não pode ser chamada de "thread-safe". Isso implica na utilização dessas bibliotecas/rotinas externas, porque você pode ter problemas, se você não tiver certeza se o código que você está usando é "thread-safe" .

3. Apresente vantagens e desvantagens de construir servidores usando um único processo com várias linhas de execução (como, p.ex. TCPmtechod) com relação a usar vários processos, cada um com uma única linha de execução (como, p. ex., TCPEchod).

A vantagem de construir um servidor com um processo e várias threads é o compartilhamento de variáveis entre as threads do mesmo processo, não precisando

compartilhar estrutura de dados.

Mas, isso depende muito da implementação do servidor, pois, pode ser mais útil compartilhar especificamente alguns dados (como em processos) do que compartilhar todos (como em thread). No uso de processos, pode-se compartilhar dados específicos, usando memória compartilhada, entre outros.

A implementação do linux não difere do ponto de vista interno uma thread de um processo (na sua criação / escalonamento), a única diferença mesmo é que a thread compartilha o mesmo segmento de dados do seu processo, ao contrário dos processos que herdam o segmento de dados.

As vantagens/desvantagens variam de acordo com a implementação / necessidade da aplicação.

4. O servidor TCPmechod implementa concorrência aparente usando uma única linha de execução. Em que circunstâncias essa abordagem é melhor que as duas apresentadas anteriormente (concorrência real com vários processos, cada um com uma única linha de execução ou concorrência real um único processo com várias linhas de execução) ?

Quando o computador apresentar poucos recursos de software, incluindo o sistema operacional ter poucos recursos disponíveis. O consumo de hardware por ser tipo de implementação é bem menor, cerca de 10% das outras implementações.

5. Um servidor que usa uma única linha de execução pode deixar de atender algum cliente enquanto repetidamente atende a outro cliente? E servidores que usam vários processos com uma única linha de execução, ou um processo com várias linhas de execução? Justifique.

Sim, se o processo ficar bloqueado em uma linha de execução esperando algum retorno, ele pode bloquear a resposta para outro cliente.

Um processo com várias linhas de execução pode deixar de atender o cliente caso o sistema operacional bloqueie recursos para uma nova linha de execução, mesmo caso para vários processos filhos.

6. Monte experimentos que permitam comparar os 3 servidores concorrentes apresentados (TCPechod, TCPmtechod e TCPmechod) e determine:

a) O número máximo de clientes que cada um é capaz de atender até recusar serviço

TCPmechod_m : enviados: 3977 respostas: 3949

TCPmtechod_m : enviados: 3974 respostas: 3951

TCPEchod_m : enviados: 1514 respostas: 580

b) Qual executa mais rapidamente?

TCPmtechod_m : 0.005036

TCPmechod_m : 0.006090

TCPEchod_m : 0.078531

c) Como o tempo de execução varia em função do número de conexões concorrentes?

Quanto mais conexões, mais requisições o servidor terá que atender, sendo via threads ou processos, consumindo então mais recursos do sistema operacional. O

sistema operacional pode bloquear recursos ocasionando a demora na resposta para o cliente.

Referências

<http://www.llnl.gov/computing/tutorials/pthreads/> - acessado dia 21/09/2005 at 09:50am
Man pages linux

Anexo A

10.0.0.1

Linux 2.6.11.4-21.9-default (geeko@buildhost) (gcc 3.3.5 20050117) #1 1CPU [cesarkallas]

Memory:	Total	Used	Free	Shared	Buffers
Mem:	499116	488692	10424	0	48536
Swap:	160640	0	160640		

Bootup: Wed Sep 21 21:07:30 2005 Load average: 0.25 0.60 0.72 1/89 27110

—

10.0.0.10

Linux 2.6.8.1 (root@tree) (gcc 3.3.4) #1 1CPU [opensrc.(none)]

Memory:	Total	Used	Free	Shared	Buffers
Mem:	191652	189596	2056	0	49104
Swap:	305224	5212	300012		

Bootup: Sun Sep 11 22:38:10 2005 Load average: 0.00 0.00 0.00 1/81 17234

Local: Minha casa

Anexo B e C

Veja os arquivos no diretório, as sessões estão com a extensão ".sessao".