UFOP-DECOM-BCC423 Nº 03/2021

Ouro Preto, 10 de Fevereiro de 2021

4º uTP

1ª tarefa:. Lembra-se do último TP? Pois é, o programa fib.c tinha o código fib.c https://gist.github.com/ecarrara/5174082 , e você, entendeu por que e como o programador usou a instrução FORK, como usou "*pipe*" e você sabe a diferença entre threads e fork, certo? Sabe também, que não há necessidade (nem tem como) usar *pipe* para fazer a comunicação entre dois threads, certo?

Pois bem, veja que *pipe* permite a "Comunicação Inter Processo" (IPC- Inter process communication) e ele foi idealizado para isto. Voltando ao nosso último TP, cada processo criado tinha um espaço de endereço único e o *pipe* permitiu que os dois processos se comunicassem. Porém, foi necessário saber o PID (*Process Identification*) dos processos. Não foi mesmo?

Obviamente, os números de PID dos processos executando em um sistema operacional (SO) pertencem exclusivamente à aquele SO. Em outras palavras, o mecanismo de *pipe* não dá para ser usado para comunicação entre processos que executam em sistemas operacionais distintos. Então, por exemplo, para comunicar do seu computador com um processo que executa no servidor da UFOP, não dá para usar "pipe".

Mais e aí, como resolver isto? Como trocar mensagens (IPC por troca de mensagens) entre computadores (hosts) distintos? Bom, por que não enviar mensagens de computadores diferentes usando o número de porta e o número IP? O número IP permitirá comunicar com qualquer computador na Internet e o número da porta permitirá que processos (um processo é um programa, um código, em execução) sejam associados a este número.

E como isto funciona? Bem, veja a biblioteca "sockets". Esta biblioteca e sua API é o padrão "de fato" da Internet. Antigamente, tínhamos uma saudável "competição" entre o padrão sockets da Universidade de Berkeley e TLI (Transport Layer Interface), presente da especificação do Unix System V da At&T. Ambas são APIs da camada de transporte de redes de computadores. Mas, o que eles fazem:

1- IPC entre *hosts* distintos. Cada *host*, normalmente, tem um SO distinto;
2- Por causa da pilha de protocolos de redes (no caso, TCP-IP), a localização geográfica dos computadores e seus detalhes de hardware e software (tipo de rede, tipo de SO) são abstraídos, isto é, você não precisa saber onde o computador está localizado geograficamente (pode ser seu próprio computador ou no Japão, por exemplo), não precisa saber se o computador usa uma rede de fibra-ótica, se é cabeado, velocidade, etc..

e nem precisa saber qual SO (Unix, Linux, Windows, MAC OS, etc...) está executando na máquina destino.

Pois bem.. falado tudo isto.. você deverá modificar o programa fib.c para usar IPC por troca de mensagens sockets em processos ou máquinas distintas

Entregáveis:

a. programa fib.c RODANDO (mostrar no vídeo) e modificado conforme solicitado neste 4º uTP;
b. vídeo mostrando o funcionamento no máximo 3 minutos. Mostre que os processos estão sendo executados distintamente;
c. arquivo em formato PDF de texto explicando o que foi feito;
d. Nste TP você deverá fazer o que foi solicitado, explicando o que difere SOCKETS, PIPE e por que não dá para usar PIPE com threadas. Compare SOCKETS, PIPE e comunicação inter-threads. Quais as diferenças entre eles? Me explique como o programa funciona, seus resultados e as técnicas de concorrência usadas.

10-fev-2021
(texto oculto presente no formato original)