





## Teste de Conhecimento

avalie sua aprendizagem

PROGRAMAÇÃO DE SOFTWARE BÁSICO CCT0029_A10_20210110137_V1	Lupa Calc.  ○ ⊕  □
Aluno: DOUGLAS MATOS DA SILVA	Matr.: 202101110137
DBC.: PROG.SOFTW.BÁSICO	2022.1 EAD (G) / EX
Prezado (a) Aluno(a),	
Você fará agora seu <u>TESTE DE CONHECIMENTO!</u> Lembre-se que este exercício é opcional, mas não composto de questões de múltipla escolha.  Após responde cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. A	
questões que será usado na sua AV e AVS.	
1. Um processo pode ser	
<ul> <li> <ul> <li>de thread única ou de multithread.</li> <li>somente de multithread.</li> </ul> </li> </ul>	
<ul> <li>de multithread só se for em vários núcleos.</li> <li>somente de thread única.</li> </ul>	
o de dois threads no máximo.	
Explicação:	
Um processo pode ter uma ou várias threads, mesmo em um processador de núcleo ú execução em um único núcleo pode ter condições de corrida, assim como um processo processos de corrida, assim como um processo proc	único. Um processo com vários threads em o com vários threads em vários núcleos.
2. Considerando o conceito de threads na linguagem C, pode-se definir MUTEX como:	
◆ ● forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads	
oforma de mutar uma thread, para que haja evolução dos processos forma de garantir acesso multiplo à memória pelas threads	
oforma de exclusão da thread forma de garantir que uma thread aguarde até que outras sejam finalizadas	
Explicação:	
Explicação: forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais	
3. São, respectivamente, uma vantagem e uma desvantagem de usar for	k e não threads:
<ul> <li></li></ul>	r mais memória RAM
<ul> <li>Não se preocupar em programar os detalhes de sincronização / Gera</li> <li>Gerar threads em diferentes processos / Usar mais memória RAM</li> </ul>	
Gerar threads em diferentes processos / Ter que se preocupar em p	rogramar os detalhes de
sincronização	
<ul> <li>Usar menos memória RAM / Gerar threads só no mesmo processo</li> </ul>	
Explicação:  O uso da chamada fork resulta na criação de um processo filho que nada compartilha	
Explicação:	em usar técnicas de sincronização, como
Espicação:  O uso da chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preccupar	em usar técnicas de sincronização, como
Explicação:  O uso da chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há	em usar técnicas de sincronização, como
Explicação:  O uso de chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:	em usar técnicas de sincronização, como
Explicação:  O uso da chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  O forma de garantir a casso exclusivo a um recurso compartilhado por duas ou mais threads Forma de garantir acesso multiplo à memória pelos threads	em usar técnicas de sincronização, como
Explicação:  O uso de chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:    forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   forma de clonar um processo   forma de donar um processo   forma de exclusão de uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como
Explicação:  O uso da chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  o forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads  of forma de darantir que um processo  forma de garantir que uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como
Explicação:  O uso da chemada fortr resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de garantir cacesso multiplo à memória pelas threads  forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de exclusão de uma thread	em usar técnicas de sincronização, como
Explicação:  O uso da chemada fortr resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de garantir cacesso multiplo à memória pelas threads  forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de exclusão de uma thread	em usar técnicas de sincronização, como
Explicação:  O uso da charmada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads  forma de conar um processo  forma de agrantir que uma thread aguarde até que outras sejam finalizadas  branda de conar um processo  Explicação:  forma de clonar um processo  forma de clonar um processo	em usar técnicas de sincronização, como
Explicação:  O uso da chamada forir resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o forir, gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  o forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de exclusão de uma thread aguarde até que outras sejam finalizadas  Explicação: forma de clonar um processo  Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:	em usar técnicas de sincronização, como
Deplicação:  O uso da chamada forir resulta na criação de um processo filho que nada compartilhe número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o forir gera um novo processo, há  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de exclusão de uma thread  Explicação:  forma de cionar um processo  Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:  forma de garantir acesso multiplo à memória pelas threads  forma de garantir acesso multiplo à memória pelas threads	em usar técnicas de sincronização, como
Duso da chamada fork resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há Considerando o conceito de processo, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de danar um processo forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de exclusão de uma thread aguarde até que outras sejam finalizadas  Explicação: forma de cionar um processo  considerando o conceito de threads, utilizando a linguagem C, pode-se definir 30IN como: forma de garantir acesso multiplo à memória pelas threads forma de aguarde várias threads em uma única  forma de agrupar várias threads em uma única forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir que uma thread segurar várias threads on uma única	em usar técnicas de sincronização, como
Dupo da chamada fortr resulta na criação de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de garantir caesso multiplo à memória pelas threads  forma de garantir que uma thread aguarde até que outras sejam finalizadas  forma de conar um processo  Spilicação:  forma de conar um processo  considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:  forma de garantir cue su multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de garantir acesso multiplo à memória pelas threads  forma de prantir que uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como
Dupo da chamada fork resulta na crisção de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:    forma de garantir acesso multiplo à memória pelas threads   forma de garantir acesso multiplo à memória pelas threads   forma de exclusão de uma thread squarde até que outras sejam finalizadas   forma de exclusão de uma thread    Fapilicação:   forma de conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   forma de exclusão de uma thread linguagem C, pode-se definir JOIN como:   forma de exclusão da thread   forma de acyupar várias threads em uma única   forma de agrantir acesso multiplo à memória pelas threads   forma de agrupar várias threads em uma única   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma transcentir que uma tran	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir XOIN como:  forma de garantir que uma thread aguarde até que outras sejam finalizadas  considerando o conceito de threads. Vultizando a linguagem C, pode-se definir XOIN como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads  forma de garantir que uma thread aguarde até que outras sejam finalizadas  considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:  forma de conar um processo  forma de conar um processo  considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:  forma de conar um processo  considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:  forma de garantir acesso multiplo à memória pelas threads  forma de exclusão da thread  forma de garantir acesso multiplo à memória pelas threads	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir JOIN como:  forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de conceito de threads, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir acesso multiplo à memória pelas threads forma de conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:  como de conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:  forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir acesso multiplo à memória pelas threads forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir que uma thread aguarde até que outras sejam finalizadas forma de garantir que uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir XOIN como:  forma de garantir que uma thread aguarde até que outras sejam finalizadas  considerando o conceito de threads. Vultizando a linguagem C, pode-se definir XOIN como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads  forma de garantir que uma thread aguarde até que outras sejam finalizadas  considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:  forma de conar um processo  forma de conar um processo  considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:  forma de conar um processo  considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:  forma de garantir acesso multiplo à memória pelas threads  forma de exclusão da thread  forma de garantir acesso multiplo à memória pelas threads	em usar técnicas de sincronização, como
Duso da chamada fork resulta na criação de um processo filho que nada compartilhe número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há como compartilhado por duas ou mais threads.  Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:  forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads forma de garantir que uma thread aguarde até que outras sejam finalizadas.  forma de exclusão de uma thread de la linguagem C, pode-se definir JOIN como:  forma de cionar um processo  Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:  forma de garantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de agrantir acesso multiplo à memória pelas threads  forma de garantir acesso multiplo à memória pelas threads  forma de garantir que uma thread aguarde até que outras sejam finalizadas  forma de garantir que uma thread aguarde até que outras sejam finalizadas  explicação:  forma de garantir que uma thread aguarde até que outras sejam finalizadas  cuantas vezes é impressa a frase "Ola Mundo!" no programa abaixo?  #include  #include  #include  #include	em usar técnicas de sincronização, como
Duso da chamada fortr resulta na crisção de um processo filho que nada compartilha número de identificação (PID). Desta forma, o programador não precisa se preocupar as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:    forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   forma de garantir caseso multiplo à meméria pelas threads   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de exclusão de uma thread   Fapilicação:   forma de conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   forma de agrantir acesso multiplo à memória pelas threads   forma de agrantir acesso multiplo à memória pelas threads   forma de agrantir acesso multiplo à memória pelas threads   forma de agrantir acesso multiplo à memória pelas threads   forma de garantir acesso multiplo à memória pelas threads   forma de garantir acesso multiplo à memória pelas threads   forma de garantir acesso exclusivo a um recurso compartilhado por duas ou mais threads   forma de garantir que uma thread aguarde até que outras sejam finalizadas   forma de garantir que uma thread aguarde até que outras sejam finalizadas   Capulatas vezes é impressa a frase "Ola Mundo!" no programa abalxo?   #include	em usar técnicas de sincronização, como
Considerando conceito de processos, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Forma de garantir que uma thread aguarde até que outras sejam finalizadas   Forma de exclusão de uma tiread aguarde até que outras sejam finalizadas   Forma de garantir acesso multiplo à memória pelas threads   Forma de exclusão de uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos conceito de uma turead compartilhado por duas ou mais threads forma de garantir que uma thread aguarde até que outras sejam finalizadas   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir XOIN como:   Considerando o conceito de threads,	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos     Considerando o conceito de uma thread aguardo eté que outras sejam finalizadas     Considerando o conceito de uma thread     Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizand	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir 70FK como:   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de conar um processo   Torma de conar um processo   Torma de conar um processo   Torma de garantir que uma thread da linguagem C, pode-se definir 30FN como:   Torma de garantir que uma thread se uma mínica     Torma de garantir acesso multiplo à memória pelas threads   Torma de garantir acesso multiplo à memória pelas threads   Torma de garantir acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de processos, utilizando a linguagem C, pode-se definir FORK como:   Considerando o conceito de uma turcedo compartilhado por duas ou mais threads     Considerando o conceito de uma turcedo     Considerando o conceito de uma turcedo     Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:   Considerando o conceito de threads, utiliza	em usar técnicas de sincronização, como
Considerando o conceito de processos, utilizando a linguagem C, pode-se definir 70FK como:   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de conar um processo   Torma de conar um processo   Torma de conar um processo   Torma de garantir que uma thread da linguagem C, pode-se definir 30FN como:   Torma de garantir que uma thread se uma mínica     Torma de garantir acesso multiplo à memória pelas threads   Torma de garantir acesso multiplo à memória pelas threads   Torma de garantir acesso exclusivo a um recurso compartilhado por duas ou mais threads   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas   Torma de garantir que uma thread aguarde até que outras sejam finalizadas	em usar técnicas de sincronização, como

