



# Teste de Conhecimento

avalie sua aprendizagem

PROGRAMAÇÃO DE SOFTWARE BÁSICO  
CCT9829\_A10\_202101110137\_V1

Lupa Calc.

Aluno: DOUGLAS MATOS DA SILVA  
Disc.: PROG.SOFTW.BÁSICOMatr.: 202101110137  
2022.1 EAD (G) / EX

Prezado (s) Aluno(a),

Você fará agora seu **TESTE DE CONHECIMENTO!** Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responder cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

1. Um processo pode ser \_\_\_\_\_.

- ✓ ☒ de thread única ou de multithread.  
☐ somente de multithread.  
☐ de multithread só se for em vários núcleos.  
☐ somente de thread única.  
☐ de dois threads no máximo.

Explicação:

Um processo pode ter uma ou várias threads, mesmo em um processador de núcleo único. Um processo com várias threads em execução em um único núcleo pode ter condições de corrida, assim como um processo com várias 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  
☐ forma de mutar uma thread, para que haja evolução dos processos  
☐ forma de garantir acesso múltiplo à memória pelas threads  
☐ forma de exclusão da thread  
☐ forma de garantir que uma thread aguarde até que outras sejam finalizadas

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 fork e não threads:

- ✓ ☒ Não se preocupar em programar os detalhes de sincronização / Usar mais memória RAM  
☐ Não se preocupar em programar os detalhes de sincronização / Gerar threads só no mesmo processo  
☐ Gerar threads em diferentes processos / Usar mais memória RAM  
☐ Gerar threads em diferentes processos / Ter que se preocupar em programar os detalhes de sincronização  
☐ Usar menos memória RAM / Gerar threads só no mesmo processo

Explicação:

O uso da chamada fork resulta na criação de um processo filho que nada compartilha com o processo pai, a não ser o seu número de identificação (PID). Desta forma, o programador não precisa se preocupar em usar técnicas de sincronização, como as usadas no caso de threads. Por outro lado, como o fork gera um novo processo, há maior consumo de memória RAM.

4. 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 acesso múltiplo à memória pelas threads  
✓ ☒ forma de clonar um processo  
☐ forma de garantir que uma thread aguarde até que outras sejam finalizadas  
☐ forma de exclusão de uma thread

Explicação:

forma de clonar um processo

5. Considerando o conceito de threads, utilizando a linguagem C, pode-se definir JOIN como:

- ☐ forma de garantir acesso múltiplo à memória pelas threads  
☐ forma de exclusão da thread  
☐ forma de agrupar várias threads em uma única  
✓ ☒ forma de garantir que uma thread aguarde até que outras sejam finalizadas  
☐ forma de garantir o acesso exclusivo a um recurso compartilhado por duas ou mais threads

Explicação:

forma de garantir que uma thread aguarde até que outras sejam finalizadas

6. Quantas vezes é impressa a frase "Ola Mundo!" no programa abaixo?

```
#include
#include
#include
int main()
{
    fork();
    printf("Ola Mundo!\n");
    return 0;
}
```

- ✓ ☒ Duas  
☐ Seis  
☐ Uma  
☐ Quatro  
☐ Nenhuma

Explicação:

O número de vezes que "Ola Mundo!" é impresso é igual ao número de processos criados. Número total de processos =  $2^n$ , em que  $n$  é o número de chamadas ao fork. Como  $n = 1$ ,  $2^1 = 2$ .

 Não Respondida  Não Gravada  Gravada

Exercício iniciado em 20/04/2022 09:32:11.