

**Dupla: Pedro Anicio Dornelas Moraes
Leandro Balta Braga**

1 - O retorno da função é o ID da thread recém-criada ou um valor negativo em caso de erro, sendo que esses IDs são armazenados nas variáveis tid1 e tid2 conforme o código fornecido.

2 - A distinção fundamental reside no método pelo qual a função fork() cria um novo processo como uma cópia idêntica do processo "pai", mas com um ID diferente, enquanto a função clone() gera threads adicionais em um processo existente, compartilhando tanto o espaço de memória quanto o ID do processo "pai".

3 - O comportamento pode variar dependendo da função utilizada para gerar as threads, podendo ocorrer a terminação total do programa ou o aguardo da finalização das threads para que o programa seja encerrado de acordo com qual função voce utiliza.

4 - Chamar a função fork() a partir de uma thread em execução pode ocasionar em problemas de sincronização, pois a fork() é destinada à criação de um novo processo ao invés de uma nova thread, podendo resultar em erro.

5 - Quando se utiliza clone() a partir de uma thread existente é criando uma nova thread dentro do mesmo processo, compartilhando de recursos e espaço de endereço.

6 - No momento em que uma thread invoca uma das funções da família exec() durante sua execução a thread atual é completamente descartada e substituída pelo novo código, que começa a ser executado desde o início.

7 – Se a função principal encerrar sua execução mas as threads geradas por pthread_create() ainda estiverem em execução, é que ocorra

vazamento de memória com a continuação da execução, ou que elas se encerrem em sincronismo com a função principal.

8 - Quando a função `fork()` é chamada, apenas a thread que a invocou é duplicada no novo processo, não existindo garantia de que as outras threads do processo original também serão copiadas. Isso pode gerar vários erros no programa, pois a função `fork()` foi projetada para duplicar um processo, não uma thread.

9 - Caso uma thread criada com `pthread_create()` invoque a função `clone()` durante sua execução, será gerado uma nova thread dentro do mesmo processo usando o sistema de threads do Linux. Isso contrasta com a função `fork()`, que cria um novo processo, enquanto a `clone()` gera uma nova thread dentro do mesmo processo, implementando, assim, uma nova thread.

10 - Quando a thread atual invoca uma das funções da família `exec()`, a thread é substituída pelo novo programa especificado pela função `exec()`. Em outras palavras, o novo programa começa a ser executado no contexto da thread atual, e a thread original é encerrada.