

Lista 12 - Programação II

Aluna: Raissa Caroline Cavalcanti

1: O que o comando "run" faz e por que ele é tão importante no início da depuração?

-> O comando "run" serve para começar a executar o programa dentro do GDB, o que é essencial para acompanhar o que está acontecendo com o código. Isso é útil para identificar onde as falhas acontecem, como quando ocorre um "segmentation fault".

2: Qual é o propósito do comando "bt" e o que ele mostra sobre o local do erro?

-> Mostra a pilha de chamadas, ou seja, as funções que foram chamadas até o momento em que o erro apareceu. Isso ajuda a entender o caminho que o programa percorreu até encontrar o problema, facilitando a localização da causa do erro.

3: Que partes do código aparecem ao redor do ponto onde o erro ocorre? Qual comportamento estranho pode ser percebido?

-> Geralmente, a função que causou o erro e algumas linhas de código próximas. Comportamentos suspeitos podem incluir coisas como tentar acessar ponteiros que não foram inicializados ou usar índices fora do limite de um array, o que costuma causar problemas.

4: Qual é a função dos comandos "up" e "down"? E qual a diferença entre os comandos "info args" e "info locals"? O que os valores que você vê podem dizer sobre a causa do erro?

-> Permitem que você navegue pela pilha de chamadas, olhando para diferentes funções e variáveis em contextos diferentes. O comando "info args" mostra os argumentos da função atual, enquanto o "info locals" exibe as variáveis locais. Se encontrar valores inesperados, isso é um indício de que algo está acessando a memória de forma errada.

5: O que o comando "next" faz e como ele ajuda a identificar o local exato do erro?

-> **Resposta:** O "next" faz o programa avançar para a próxima linha sem entrar em funções chamadas, o que permite ver se o problema está acontecendo na própria linha de código ou dentro de uma função chamada.

6: Qual é a diferença entre os comandos "step" e "next"? O endereço do buffer dentro de uma função é o mesmo fora dela?

-> O "step" entra dentro de funções chamadas para acompanhar a execução delas passo a passo, enquanto o "next" simplesmente vai para a próxima linha de código. O endereço do buffer costuma ser o mesmo, porque é a mesma variável sendo acessada, a menos que esteja sendo passada por referência.

7: Como o comando "watch" ajuda a acompanhar as mudanças em uma variável? O que acontece no depurador quando o valor de "j" muda?

->**Resposta:** O comando "watch" é útil porque permite observar de perto uma variável e seja notificado assim que o valor dela mudar. Quando o valor de "j" se altera, o depurador pausa a execução, e você pode ver o que está acontecendo com o programa naquele exato momento.

8: O que provavelmente está causando o erro no código?

->O erro provavelmente está acontecendo por causa de um acesso incorreto à memória, como tentar usar um ponteiro que não foi inicializado ou acessar uma posição de array que está fora do limite.