

Problemas com Buffer na Leitura em C

Prof. Dr. Anderson Bessa da Costa
Algoritmos e Programação 2
FACOM, Universidade Federal de Mato Grosso do Sul

1 Introdução

Na leitura de dados utilizando a função `scanf()` por vezes problemas são causados pela manipulação do buffer de entrada. Neste material será mostrado como esse problema aparece e uma solução simples para corrigi-lo.

2 Motivação

Considere o código abaixo:

```
1 #include <stdio.h>
2
3 int main() {
4     int a;
5     char b;
6
7     printf("Entre com um inteiro: ");
8     scanf("%d", &a);
9
10    printf("Entre com um caractere: ");
11    scanf("%c", &b);
12
13    printf("\nO numero inteiro lido foi %d. O caractere lido foi %c.", a, b);
14    return 0;
15 }
```

Considere a seguinte saída:

```
Entre com um inteiro: 3
Entre com um caractere:
O numero inteiro lido foi 3. O caractere lido foi
.
```

O código apresentado solicita ao usuário a entrada de um número inteiro e logo em seguida a entrada de um caractere. O problema é que ao entrarmos com o número inteiro e apertarmos ENTER, não é solicitada a entrada do caractere.

3 Solução

A função `scanf()` armazena em um buffer o conteúdo digitado pelo usuário e armazena também a tecla ENTER utilizada pelo usuário para encerrar a entrada de dados. Para que o buffer seja esvaziado depois da atribuição do conteúdo à variável, utiliza-se `scanf("%c", &b);`.

3.1 Exemplo

O código abaixo implementa a solução.

```
#include <stdio.h>

int main() {
    int a;
    char b;

    printf("Entre com um inteiro: ");
    scanf("%d%c", &a);

    printf("Entre com um caractere: ");
    scanf("%c%c", &b);

    printf("\nO numero inteiro lido foi %d. O caractere lido foi %c.", a, b);

    return 0;
}
```

Considere a seguinte saída:

```
Entre com um inteiro: 3
Entre com um caractere: a
```

```
O numero inteiro lido foi 3. O caractere lido foi a.
```

Adicionar `%c` esvaziou o buffer e permitiu a leitura correta dos dados. É recomendado que seja adicionado a sequência em todas as chamadas da função `scanf()` para que o buffer seja efetivamente esvaziado antes de uma nova leitura de dados.