

自作出力関数 snapshot() の解説

snapshot 関数は、可変長引数を受け取り、現在の日時と共にフォーマットされたメッセージを標準出力に表示するものです。

使用しているライブラリ:

<stdio.h>: 標準入出力を行うためのライブラリで、fprintf や snprintf などの関数を提供します。

<stdarg.h>: 可変長引数进行处理するためのライブラリで、va_list、va_start、va_end などのマクロを提供します。

<time.h>: 時刻や日付を扱うためのライブラリで、time、localtime、strftime などの関数を提供します。

バッファオーバーフローの防止:

vsnprintf や snprintf を使用して、指定したサイズ内に収まるように出力を制限しています。これにより、バッファオーバーフローを防止しています。

エラーチェック: vsnprintf や snprintf の戻り値を確認し、出力がバッファサイズを超えた場合やエラーが発生した場合に適切な処理を行っています。

可変長引数の処理: va_list、va_start、va_end を使用して可変長引数を適切に処理しています。これにより、柔軟な引数の受け渡しが可能となっています。

```
1 #include <stdio.h>
2 #include <stdarg.h>
3 #include <time.h>
4
5 #define MAX_SIZE 4096
6
7 extern int snapshot(const char *fmt, ...)
8 {
9     char tstr[MAX_SIZE];
10    char buff[MAX_SIZE];
11    char message[MAX_SIZE];
12    time_t ct;
13    struct tm *lst;
14    va_list ap;
15    int ret;
16
17    // 現在時刻を取得
18    time(&ct);
19    lst = localtime(&ct);
20    strftime(tstr, sizeof(tstr), "%Y/%m/%d %H:%M:%S", lst);
21
22    // 可変長引数の処理開始
23    va_start(ap, fmt);
24
25    // フォーマットされたメッセージを作成
26    ret = vsnprintf(message, sizeof(message), fmt, ap);
27    if (ret < 0)
28    {
29        // エラー処理: vsnprintf が失敗した場合
30        va_end(ap);
31        return -1;
32    }
33    else if (ret >= sizeof(message))
34    {
35        // エラー処理: 出力がバッファサイズを超えた場合
36        va_end(ap);
37        return -2;
38    }
39
40    // タイムスタンプとメッセージを結合
41    ret = snprintf(buff, sizeof(buff), "| %s\n %s", tstr, message);
42    if (ret < 0)
43    {
44        // エラー処理: snprintf が失敗した場合
45        va_end(ap);
46        return -3;
47    }
48    else if (ret >= sizeof(buff))
49    {
50        // エラー処理: 出力がバッファサイズを超えた場合
51        va_end(ap);
52        return -4;
53    }
54
55    // 可変長引数の処理終了
56    va_end(ap);
57
58    // 標準出力に出力
59    fprintf(stdout, "%s", buff);
60
61    return 0;
62 }
```