

# GDB-调试点

朱德森

2021 年 10 月 22 日

## Contents

1	背景	2
2	常用命令	2
3	举例使用	2
4	参考资料	4

## 1 背景

在进行程序的调试时，经常需要反复运行程序至问题所在，而有时需要等待大量时间运行至问题处，若不能成功定位到问题，就需要重新运行，消耗大量不必要的时间。

GDB 的调试点针对该问题提供了解决方案，在调试过程中即使没能定位到问题，也只需要花费一次时间运行至问题处。

GDB 调试点(checkpoint) 的原理是复制进程的方式，当程序运行到问题点处时，通过checkpoint 命令将当前调试进程复制一份。那么当第一个进程执行到问题点之后时，可以切换至复制出的进程，此时复制出的进程依然停留在问题点处，可以再一次进行调试，而不需要将程序重头运行一遍。

## 2 常用命令

- checkpoint : 生成新的调试点
- restart [num] : 跳转到调试点[num]
- delete checkpoint [num] : 删除调试点[num]
- info checkpoint : 查看当前所有调试点信息

## 3 举例使用

下面提供了一个工程例子进行说明，如图4 所示，

本段代码主要是模拟一个百分数的累加，也可以认为是程序进度的一个指示。为了调试，在AddPercent 函数中人为地引入了一个bug，导致在主程序中百分比的累加出现不连续，下面将利用这段程序来演示checkpoint 调试点的使用。

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 void AddPercent(int* percent) {
6     (*percent)++;
7     // This code acts as bug.
8     if (*percent == 30) {
9         (*percent)++;
10    }
11    return;
12 }
13 int main(int argc, char* argv[]) {
14     int percent = 0;
15     for (int i = 0; i < 100; i++) {
16         AddPercent(&percent);
17         /*printf("Current percent = %d\\%\\n", percent);*/
18     }
19     printf("Final percent = %d\\%\\n", percent);
20
21     return 0;
22 }

```

Figure 1: 代码例子

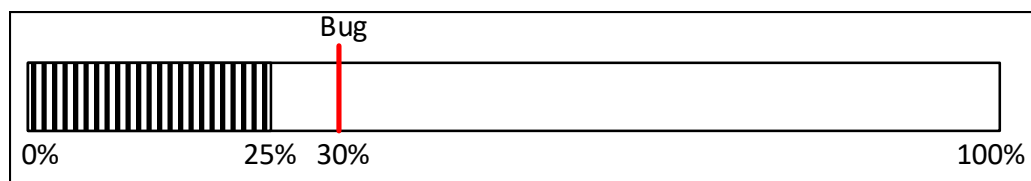


Figure 2: 代码例子

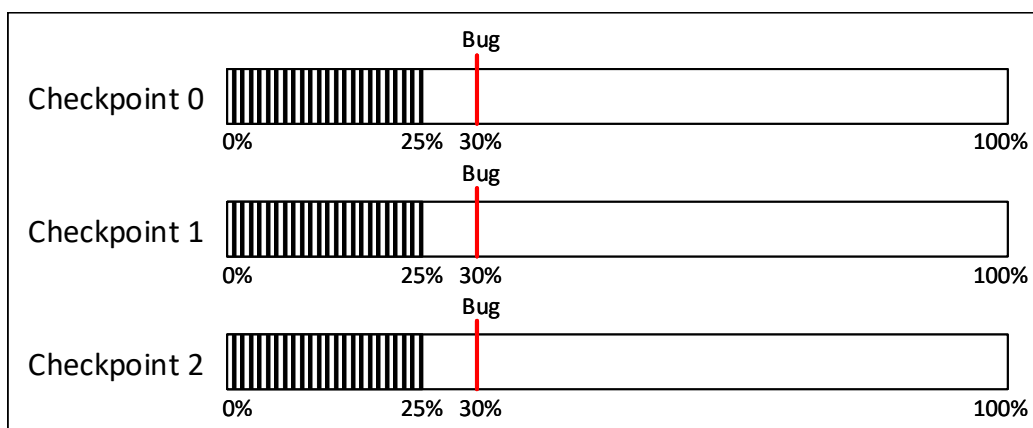


Figure 3: 代码例子

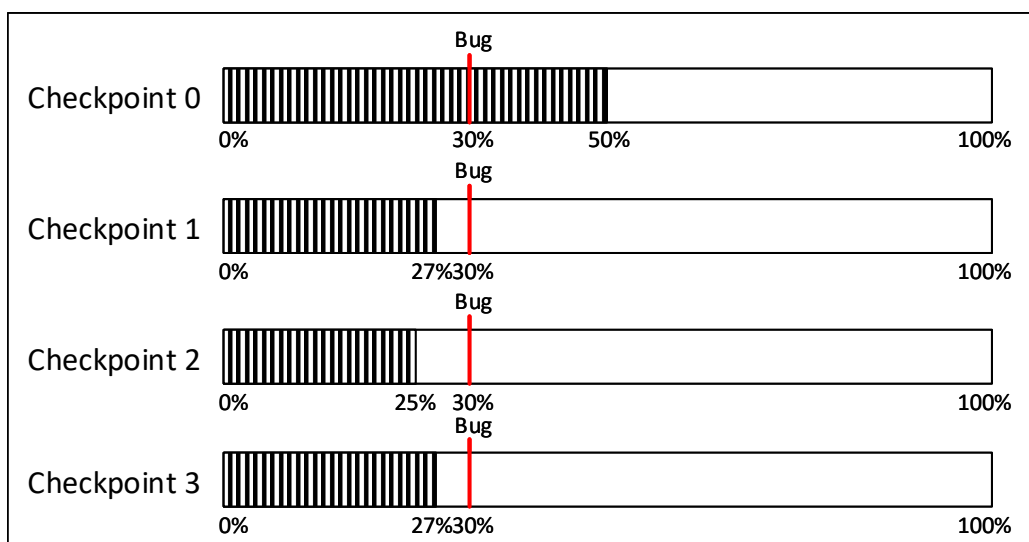


Figure 4: 代码例子

## 4 参考资料

- gdb使用小技巧-保存调试点现场(<https://www.cnblogs.com/qinghaowusu/p/13994017.html>)