

利用 Valgrind 对程序性能调优

前言

valgrind 不仅可以用来检测内存泄露，还可以用来获取函数调用关系，找出性能瓶颈！

Example:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void f1()
{
    int i;
    for (i = 0; i < 30000; i++)
    {
        printf("hello world!\n");
    }
}

void f2()
{
    int i;
    int *p;
    for (i = 0; i < 30000; i++)
    {
        p = malloc(sizeof(int));
        *p = 10;
        free(p);
    }
}

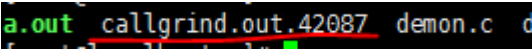
void f3()
{
    char str[5];
    int i = 0;
    for (i = 0; i < 30000; i++)
    {
        memset(str, 0, 5);
    }
}

int main()
{
    f1();
    f2();
    f3();

    return 0;
}
```

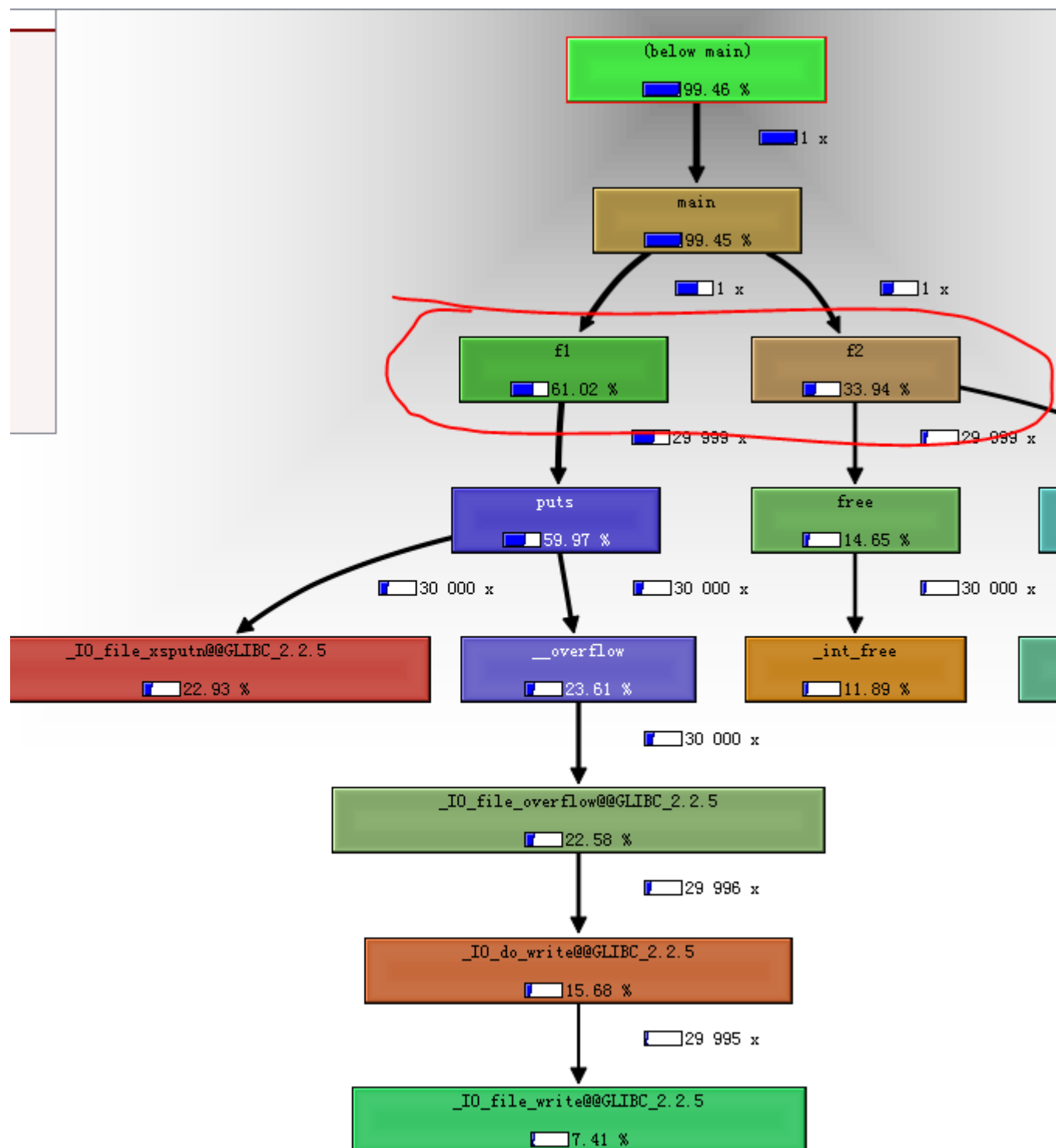
1)编译 gcc -g demon.c(一定要加上-g 选项)

2) 运行 : valgrind --tool=callgrind --dump-instr=yes --trace-jump=yes ./a.out

3)得到 out 文件: , 后面的数字表示进程 id

4)对输出结果进行分析

把 out 文件拷贝到 windows 下, 打开分析软件 kcachegrind.exe, 对 out 文件进行分析



从图中可以看出最耗时的操作在函数 f1,f2 次之！