

程序执行的时空测量方法研究

黎远松

(四川理工学院计算机学院, 四川 自贡 643000)

摘要:程序执行时所花费的时间和内存空间是评价一个程序性能优劣的一个重要指标。理论上,可以借助于数学这个强有力的工具估算出程序的时间复杂度和空间复杂度,然后用这两个标准来度量程序的效率;实际上,在源代码在线评测系统这样的实际系统中无法使用。文章对程序执行时所花费的时间和内存空间的测量方法进行大量深入的研究,基于 DOS 平台,提出了计时法,利用 clock 和 coreleft 测量程序执行的时间和空间;基于 Windows 平台,利用 Get Process Times 和 Get Process MemoryInfo 测量程序执行的时间和空间,在四川理工 ACM 中得到应用,实践证明这些方法正确、有效。

关键词:程序;时间;空间;测量;方法

中图分类号:TP393.6

文献标志码:A

在源代码在线评测系统的研究过程中发现,测量进程执行时所花费的时间和所占用的内存空间是一个亟待解决的重要问题。利用 API 函数 Get Tick Count() 获取进程的执行时间^[1],在多任务环境里是不准确的,负载的变化对实验结果影响明显。

Windows 2000 引入了新的进程控制原语——作业对象。作业对象是可以作为单个实体管理的一个或多个进程的集合。用户可以设置整个执行时间和处理器时间的分配、以及控制作业对象的调度选项^[2],但编程实现略显复杂。

针对这些问题,本文进行大量深入的研究,提出了以下适用不同环境的测量方法。

1 基于 DOS 平台的测量

1.1 运行时间的测量

函数 clock() 返回程序开始运行时经过的时间,用它测定运行时间,方法如下:

```
t1 = clock();
```

被测程序

```
t2 = clock();
```

```
t3 = t2 - t1;
```

在虚拟机(16M,单核 2.33Ghz,DOS7.1,BC31)中测量以下程序的实验数据见表 1。

```
for(a = 100; a < 200; a++) {
```

```
for(b = 100; b < 200; b++) {
```

```
for(c = 100; c < 200; c++) {
```

```
d = 711 - a - b - c;
```

```
if(a * b * c * d == 711000000)
```

```
}}}
```

表 1 clock 实验结果

分组情况	t3 (tick)	time (s)
第一组	320	17.582418
第二组	321	17.637363
第三组	319	17.527473
第四组	317	17.417582
第五组	319	17.587473

从表 1 中的 time 数据可以看出,被测程序的运行时间稳定在 17 s。

由于 DOS 是单任务操作系统,被测程序受环境的影

收稿日期:2013-10-09

基金项目:四川省教育厅科研项目(13ZAO125);软件工程专业综合改革项目(ZG-1202)

作者简介:黎远松(1970-),男,重庆开县人,副教授,硕士,主要从事源代码在线评测系统方面的研究,(E-mail)lys700620@yeah.net

响极小,因此,测试结果变化极小。

DOS 平台的测量方法是最基本的方法,测量的结果也是最容易理解的。但在多任务环境里,准确性难于保证^[3,4]。

1.2 内存空间的测量

函数 coreleft 返回以前未用的堆空间的大小。只用在没有分配或释放对象时这个值才代表真正可用的内存大小。获得准确的 RAM 大小需要遍历堆空间,利用它测定未用的 RAM 内存,方法如下:

```
r1 = coreleft ();
```

被测程序:

```
r2 = coreleft ();
```

```
r3 = r2 - r1;
```

实验结果见表 2。

表 2 coreleft 实验结果

分组情况	内存分配	r3 (KB)
第一组	malloc(1024)	1
第二组	malloc(2048)	2
第三组	int x[1024]	2

从表 2 中的数据可以看出,被测程序 malloc(1024) 在堆空间中分配了 1 KB 的 RAM 内存,未用的 RAM 内存空间减少 1 KB,正确,但是这种方法只能测定以前未用的堆空间的大小,第三组静态分配的内存用 coreleft 测量不出来,因此,具有很大的局限性。

2 基于 Windows 平台的测量

2.1 获取进程时间信息

选手程序执行时间的多少,是一个重要的评测指标^[2,5,6],这里利用 API 函数 Get Process Times() 获取进程的执行时间,方法为:

用父进程本身创建一个新进程(Create Process),获得 pi.handle,然后等程序运行完毕以后调用 Get Process Times 得到需要的时间(用户模式与 kernel 模式)。

通过传递给 Create Porcess 的 PROCESS_INFORMATION 参数,创建进程 Wait For Single Object(pi.h Process,INFINITE),然后再调用 Get Process Times。最后利用 File Time To System Time 函数转成 SYSTEMTIME,核心代码如下:

```
BOOL ret = Create Process ( argv [ 1 ], NULL, NULL, NULL, TRUE, 0, NULL, NULL, &si, &pi );
```

//argv[1]是选手程序的文件名

```
a = Wait For Single Object ( pi. h Process, INFINITE );
```

```
//等待选手程序执行 INFINITE
```

```
Get Process Times( pi. h Process, & Creation Time, & Exit Time, & Kernel Time, & User Time );
```

```
File Time To System Time( & Kernel Time, &kt );
```

```
File Time To System Time( &User Time, &ut );
```

```
//时间格式转换
```

```
t = ( ut. w Second + kt. w Second ) * 1000 + ut. w Milliseconds + kt. w Milliseconds; //秒转换成毫秒
```

测试环境:虚拟机,512 M,单核 2.33 Ghz, Windows Server 2000, VC++6.0。实验结果见表 3、表 4。

表 3 Get Tick Count 实验结果

分组情况	程序执行的时间(ms)
第一组	2.235
第二组	2.203
第三组	2.203
第四组	2.250
第五组	2.235

表 4 GetProcessTimes 实验结果

分组情况	程序执行的时间(ms)
第一组	2.218
第二组	2.234
第三组	2.218
第四组	2.218
第五组	2.218

从表 4 中的数据可以看出,利用函数 GetProcessTimes() 来获取进程的执行时间,在多任务环境里是准确的,负载的变化对实验结果影响不明显。

2.2 获取进程内存信息

选手程序执行时占用内存的多少,是另一个重要的评测指标^[7,9],获取进程执行时所占用的内存信息比获取进程执行时所花费的时间要困难得多。本文用父进程本身创建一个新进程(Create Process),获得 pi.handle,然后等程序运行完毕以后调用 Get Process Memory-Info 得到当前进程所使用的内存 Working Set Size,单位是字节,核心代码如下:

```
Get Process MemoryInfo( pi. h Process, &pmmc, sizeof( pmmc ) );
```

```
//获取某一个进程的内存信息。
```

```
cout << "l" << t << "l" << pmmc. Working Set Size / 1024 << "l"; //把 Byte 转换 KB
```

被测程序:

```
int x[1024];
```

```
int y[1024];
```

```
int z[1024];
```

表 5 GetProcessMemoryInfo 实验结果

分组情况	使用数组情况	memory(KB)
第一组	x	612
第二组	x + y	616
第三组	x + y + z	620

由表 5 可以看出, Get Process MemoryInfo 能正确地测量出程序所使用的内存情况, 分配单位为 4KB。

需要强调的是, Get Process Times() 和 Get Process MemoryInfo() 函数一定要位于 Wait For Single Object() 函数之后, 只有这样才能获取正确的信息。

3 结束语

基本 DOS 平台的测量方法简单、有效, 在算法的事后分析时得到应用; 基于 Windows 平台的测量方法已应用在理工 ACM 系统中, 运行良好。对于运行时间和空间的评测和限制等, 有待进一步研究和解决。

参考文献:

- [1] 王腾, 药单霖. Online Judge 系统的设计开发[J]. 计算机应用与软件, 2006(12):78-83.
- [2] (美) Jeffrey Richter. WINDOWS 核心编程[M]. 北京: 机械工业出版社, 2008.
- [3] 张浩斌. 基于开放式云平台的开源在线评测系统设计与实现[J]. 计算机科学, 2012, 39(11A):339-346.
- [4] 李臣龙, 鲍广喜. 基于 WAMP 的在线评测系统设计与实现[J]. 电脑编程技巧与维护, 2013, 15:64-75.
- [5] 孟繁军, 刘东升, 张丽萍, 等. 程序设计基础教学策略的实践研究[J]. 内蒙古师范大学学报: 教育科学版, 2013(3):126-129.
- [6] Mickey Williams. Windows 2000 编程技术内幕[M]. 北京: 机械工业出版社, 1999.
- [7] 唐子蛟, 项菲, 符长友. 电子投票系统结果计算的误差源分析[J]. 四川理工学院学报: 自然科学版, 2013, 26(4):52-55.
- [8] 王晓, 刘凤仙, 李琦. GIS 与三维可视化在校园地下管网线管理中的应用[J]. 四川理工学院学报: 自然科学版, 2013, 26(3):46-50.
- [9] Mihail B. The U-Kenmotsu Hypersurfaces Axiom and Six-Dimensional Hermitian Submanifolds of cayley algebra[J]. Journal of Sichuan University of Science & Engineering(Natural Science Edition), 2013, Vol.26No.3: 1-6.

Research on Approaches to Measure Program Execution Time and Space

LI Yuansong

(School of Computer Science, Sichuan University of Science & Engineering, Zigong 643000, China)

Abstract: Time and space needed for process are two essential measures of our algorithm. In theory, we calculate the time complexity and space complexity of the algorithm with the aid of mathematics firstly, and then measure the efficiency of the algorithm with these two standards. In fact, the time complexity and space complexity of the algorithm cannot be used in the actual system such as online judge system. Approaches to measure process execution time and space were deeply researched for this problem and timing based on DOS was proposed, which measures program execution timing and memory with clock and coreleft. Base on windows with get process times and get process memory Info, it had been applied in the SUSE ACM correctly and effectively.

Key words: program; time; measure; approach