

简易任务管理器

姓名：高圣迪

0. Introduction

项目名称**简易任务管理器的实现**。在之前的基础上设立了两个线程。主线程负责接收command指令。接收到show的时候把任务管理器相关信息打印出来。另一个线程中设置了一个SetTimer，让他每1S去更新一次信息。

1. Data Structures

我用了一个Map来存所有进程，key值设了PID，键值是我定义的一个结构体，具体含义如注释所示！

```
struct Process {
    WCHAR name[MAX_PATH];           //进程名
    WCHAR user_name[MAX_PATH];      //用户名
    float CPU;                       //CPU占用
    int64_t last_time = 0;           //用于记录第一次的时间，求差算CPU占用
    int64_t last_system_time = 0;
    SIZE_T memory;                  //内存占用
    ULONGLONG Ioread;                //IO读字节数
    ULONGLONG Iowrite;               //IO写字节数
    DWORD pagefault;                 //页错误数
    DWORD thread_cnt;                //线程数
    DWORD handle_cnt;                //句柄数
    DWORD GDI_cnt;                  //GDI对象数
};
```

与第一版不同的地方在IO读写，第一版用的是WORD，共四个字节，会导致overflow。

名称	PID	用户名	CPU	峰值工作集...	页面错误	句柄	线程	GDI对象	I/O 读取字节	I/O 写入字节
MfeAVSvc.exe	9180	SYSTEM	00	351,492 K	1,960,194	861	77	0	7,364,774,583	219,818,155

名称	PID	用户名	CPU	峰值工作集...	页面错误	句柄	线程	GDI对象	I/O 读取字节	I/O 写入字节
MfeAVSvc.exe	9180	SYSTEM	0.000000	359927808	1225396719	219807959	1960108	77		
ModuleCoreService.exe	9460	SYSTEM	0.000000	69824512	-1095157603	347327664	421110	46		

第二版我用了ULONGLONG，解决了这个BUG。

名称	PID	用户名	CPU	峰值工作集...	页面错误	句柄	线程	GDI对象	I/O 读取字节	I/O 写入字节
MfeAVSvc.exe	9180	SYSTEM	0.000000	359927808	7374802235	220021827	1960632	77	861	0
ModuleCoreService.exe	9460	SYSTEM	0.000000	69824512	3199810353	347327664	421127	46	1053	0

2. Functions

我一共设置了2个全局变量

```
std::mutex mt;           //互斥锁，防止更新和show对list同时操作
map<DWORD, Process> list; //key为PID，值为Process结构体
```

我一共构建了8个函数

```
map<DWORD, Process> GetPidAndProcessNameAndThreadCnt(HANDLE hSnapshot);  
void GetProcessUser(HANDLE hSnapshot);  
uint64_t convert_time_format(const FILETIME* ftime);  
void GetCpu();  
void GetMemoryAndPageFault();  
void GetIOReadAndWrite();  
void GetHandleCnt();  
void GetGdiCnt();
```

比起之前的版本，我一共改了四个地方。

1. 在 `GetPidAndProcessNameAndThreadCnt` 中遍历快照时，先是尝试查找list中有没有这个PID,有的话就更新，没有的话就像之前一样 `list.insert(make_pair(processInfo.th32ProcessID, p))`。在函数最后检查list中有没有PID是这次快照中没有的，这代表此进程已关闭，把这个 (PID, Proces) 的Pair从list中删去。
2. 在 `GetCput` 中不再遍历两次，而是遍历的时候打开这个进程，获得这个进程的所用时间和系统的时间。如果 `last_time` 和 `last_system_time` 不为0，说明这不是第一次获取时间，就可以用差值计算CPU占比。每次遍历，记录时间的时候都更新 `last_time` 和 `last_system_time`。
3. 在每次需要更新或读list的时候都 `mutex.lock()`，结束后解锁，防止同时读写同时list操作。

3. Result

```

The sum of all other process is 0.000000
The sum of process0 is 100.000000
The sum of all other process is 6.573297
The sum of process0 is 93.426704
The sum of all other process is 3.305877
The sum of process0 is 96.694122
The sum of all other process is 3.126213
The sum of process0 is 96.873787
The sum of all other process is 2.348238
The sum of process0 is 97.651764
The sum of all other process is 1.769638
The sum of process0 is 98.230362
The sum of all other process is 2.332857
The sum of process0 is 97.667145
The sum of all other process is 3.315450
The sum of process0 is 96.684547
The sum of all other process is 3.331026
The sum of process0 is 96.668976
The sum of all other process is 4.868207
The sum of process0 is 95.131790
The sum of all other process is 2.147740
The sum of process0 is 97.852257
The sum of all other process is 2.144860
The sum of process0 is 97.855141
The sum of all other process is 3.546140
The sum of process0 is 96.453857
The sum of all other process is 4.250975
The sum of process0 is 95.749023
The sum of all other process is 1.076564

```

这张图是PID为0的进程CPU占比和其他进程CPU占比的总和。可以看到这次的CPU占比比起上次更加贴合实际！

F:\TaskManager_2\TaskManager_2\Debug\TaskManager_2.exe

There are 261 processes!

Process Name	ProcessID	User Name	CPU(%)	Memory(B)	IO Read	IO Write	Page Fault	Threads	Handles	GDI
[System Process]	0	SYSTEM	97.661171	0	0	0	0	8	0	0
System	4	SYSTEM	0.000000	0	0	0	0	267	0	0
Registry	124	SYSTEM	0.000000	0	0	0	0	4	0	0
chrome.exe	260	40280	0.000000	45895680	26450590	81252598	36560	11	307	0
services.exe	412	SYSTEM	0.000000	0	0	0	0	9	0	0
smss.exe	508	SYSTEM	0.000000	0	0	0	0	2	0	0
lsass.exe	548	SYSTEM	0.000000	29683712	36290113	4828642	230689	8	2008	0
fontdrvhost.exe	604	UMFD-0	0.000000	5038080	66082	0	2588	5	32	0
svchost.exe	856	LOCAL SERVICE	0.000000	11624448	0	0	122969	10	452	0
csrss.exe	864	SYSTEM	0.000000	0	0	0	0	12	0	0
svchost.exe	916	SYSTEM	0.000000	73891840	156919518	6705152	10342199	22	1631	0
wininit.exe	972	SYSTEM	0.000000	0	0	0	0	1	0	0
SearchProtocolHost.exe	1000	SYSTEM	0.000000	16416768	1270672	248	5119	6	379	0
NUDFHost.exe	1044	LOCAL SERVICE	0.000000	22278144	5355612	616	29473	11	331	0
ShellExperienceHost.exe	1152	40280	0.000000	59645952	45084	8	17548	10	581	0
svchost.exe	1216	NETWORK SERVICE	0.000000	28520448	1560576	0	878478	12	1988	0
svchost.exe	1264	SYSTEM	0.000000	9732096	0	0	17721	3	443	0
NUDFHost.exe	1360	LOCAL SERVICE	0.000000	6635520	0	0	4117	5	216	0
svchost.exe	1524	LOCAL SERVICE	0.000000	9420800	0	0	6657	3	200	0

F:\TaskManager_2\TaskManager_2\Debug\TaskManager_2.exe

There are 272 processes!

Process Name	ProcessID	User Name	CPU(%)	Memory(B)	IO Read	IO Write	Page Fault	Threads	Handles	GDI
[System Process]	0	SYSTEM	80.392418	0	0	0	0	8	0	0
System	4	SYSTEM	0.000000	0	0	0	0	267	0	0
Registry	124	SYSTEM	0.000000	0	0	0	0	4	0	0
chrome.exe	260	40280	0.000000	45895680	26450750	81252918	36560	11	307	0
services.exe	412	SYSTEM	0.000000	0	0	0	0	9	0	0
smss.exe	508	SYSTEM	0.000000	0	0	0	0	2	0	0
lsass.exe	548	SYSTEM	0.000000	29683712	36424541	4837638	230873	10	2056	0
fontdrvhost.exe	604	UMFD-0	0.000000	5038080	66082	0	2588	5	32	0
svchost.exe	856	LOCAL SERVICE	0.000000	11624448	0	0	122979	9	449	0
csrss.exe	864	SYSTEM	0.000000	0	0	0	0	12	0	0
svchost.exe	916	SYSTEM	9.433766	73891840	157021918	6705152	10342439	22	1676	0

第一张图是我输入show之后的result，第二张图是我开了几个浏览器之后的show的result。可以看到list的总量是变化了的。

4. Reference

VS2017用SetTimer和多线程实现定时器

https://blog.csdn.net/lishun1422840684/article/details/111407171?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242

c++ 11 多线程系列----mutex

https://blog.csdn.net/chenxun_2010/article/details/49786263