系统进程统计

提示：

#include <iostream>

#include<list>

#include <stdlib.h>

#include <string.h>

using namespace std;

class task {

public:

char name[128];

int memory;

task(char\* t) : memory(0)

{

char\* b = t + 1, \* p = t + 1;

while (\*p) {

if (\*p == '"')

{

\*p = 0;

strcpy(name, b);

break;

}

p++;

}

for (b = p + 1; p[1] != 'K'; p++)

if (\*p == '"')

b = p;

for (char\* c = b + 1; c != p; c++)

if (\*c == ',')

continue;

else

memory = 10 \* memory + \*c - '0';

}

bool operator<(task& t) { return this->memory < t.memory; }

};

int main() {

list<task> q;

char buffer[128], cmd[] = "tasklist /FO CSV";

FILE\* pipe = \_popen(cmd, "r");

if (!pipe)

return -1;

fgets(buffer, 128, pipe);

while (!feof(pipe))

if (fgets(buffer, 128, pipe))

q.push\_back(task(buffer));

\_pclose(pipe);

q.sort();

for (list<task>::iterator it = q.begin(); it != q.end(); ++it)

cout << it->name << '\t' << it->memory << endl;

return 0;

}

//

通过以上代码可以获取当前计算机的进程名字（name）和内存占用量（memory）

时间可以通过记录进程开始的时刻来计算

结束的进程可以通过连续两次刷新的结果比较获得

以上方法由柴华溢和郭世祺同学共同整理完成