<https://blog.csdn.net/zyy617532750/article/details/59145693>

1.获取CPU的利用率

**[cpp]** [view plain](https://blog.csdn.net/zyy617532750/article/details/59145693) [copy](https://blog.csdn.net/zyy617532750/article/details/59145693)

1. **typedef** **struct** CPU\_PACKED         //定义一个cpu occupy的结构体
2. {
3. **char** name[20];             //定义一个char类型的数组名name有20个元素
4. unsigned **int** user;        //定义一个无符号的int类型的user
5. unsigned **int** nice;        //定义一个无符号的int类型的nice
6. unsigned **int** system;    //定义一个无符号的int类型的system
7. unsigned **int** idle;         //定义一个无符号的int类型的idle
8. unsigned **int** iowait;
9. unsigned **int** irq;
10. unsigned **int** softirq;
11. }CPU\_OCCUPY;

主程序：

**[cpp]** [view plain](https://blog.csdn.net/zyy617532750/article/details/59145693) [copy](https://blog.csdn.net/zyy617532750/article/details/59145693)

1. **double** cal\_cpuoccupy (CPU\_OCCUPY \*o, CPU\_OCCUPY \*n)
2. {
3. **double** od, nd;
4. **double** id, sd;
5. **double** cpu\_use ;
7. od = (**double**) (o->user + o->nice + o->system +o->idle+o->softirq+o->iowait+o->irq);//第一次(用户+优先级+系统+空闲)的时间再赋给od
8. nd = (**double**) (n->user + n->nice + n->system +n->idle+n->softirq+n->iowait+n->irq);//第二次(用户+优先级+系统+空闲)的时间再赋给od
10. id = (**double**) (n->idle);    //用户第一次和第二次的时间之差再赋给id
11. sd = (**double**) (o->idle) ;    //系统第一次和第二次的时间之差再赋给sd
12. **if**((nd-od) != 0)
13. cpu\_use =100.0- ((id-sd))/(nd-od)\*100.00; //((用户+系统)乖100)除(第一次和第二次的时间差)再赋给g\_cpu\_used
14. **else** cpu\_use = 0;
15. **return** cpu\_use;
16. }
18. **void** get\_cpuoccupy (CPU\_OCCUPY \*cpust)
19. {
20. **FILE** \*fd;
21. **int** n;
22. **char** buff[256];
23. CPU\_OCCUPY \*cpu\_occupy;
24. cpu\_occupy=cpust;
26. fd = fopen ("/proc/stat", "r");
27. fgets (buff, **sizeof**(buff), fd);
29. sscanf (buff, "%s %u %u %u %u %u %u %u", cpu\_occupy->name, &cpu\_occupy->user, &cpu\_occupy->nice,&cpu\_occupy->system, &cpu\_occupy->idle ,&cpu\_occupy->iowait,&cpu\_occupy->irq,&cpu\_occupy->softirq);
31. fclose(fd);
32. }
34. **double** getCpuRate()
35. {
36. CPU\_OCCUPY cpu\_stat1;
37. CPU\_OCCUPY cpu\_stat2;
38. **double** cpu;
39. get\_cpuoccupy((CPU\_OCCUPY \*)&cpu\_stat1);
40. sleep(1);
42. //第二次获取cpu使用情况
43. get\_cpuoccupy((CPU\_OCCUPY \*)&cpu\_stat2);
45. //计算cpu使用率
46. cpu = cal\_cpuoccupy ((CPU\_OCCUPY \*)&cpu\_stat1, (CPU\_OCCUPY \*)&cpu\_stat2);
48. **return** cpu;
49. }

2.获取内存的使用情况

**[cpp]** [view plain](https://blog.csdn.net/zyy617532750/article/details/59145693) [copy](https://blog.csdn.net/zyy617532750/article/details/59145693)

1. **typedef** **struct** MEM\_PACKED         //定义一个mem occupy的结构体
2. {
3. **char** name[20];      //定义一个char类型的数组名name有20个元素
4. unsigned **long** total;
5. **char** name2[20];
6. }MEM\_OCCUPY;

9. **typedef** **struct** MEM\_PACK         //定义一个mem occupy的结构体
10. {
11. **double** total,used\_rate;
12. }MEM\_PACK;

主程序：

**[cpp]** [view plain](https://blog.csdn.net/zyy617532750/article/details/59145693) [copy](https://blog.csdn.net/zyy617532750/article/details/59145693)

1. MEM\_PACK \*get\_memoccupy ()    // get RAM message
2. {
3. **FILE** \*fd;
4. **int** n;
5. **double** mem\_total,mem\_used\_rate;;
6. **char** buff[256];
7. MEM\_OCCUPY \*m=(MEM\_OCCUPY \*)malloc(**sizeof**(MEM\_OCCUPY));;
8. MEM\_PACK \*p=(MEM\_PACK \*)malloc(**sizeof**(MEM\_PACK));
9. fd = fopen ("/proc/meminfo", "r");
11. fgets (buff, **sizeof**(buff), fd);
12. sscanf (buff, "%s %lu %s\n", m->name, &m->total, m->name2);
13. mem\_total=m->total;
14. fgets (buff, **sizeof**(buff), fd);
15. sscanf (buff, "%s %lu %s\n", m->name, &m->total, m->name2);
16. mem\_used\_rate=(1-m->total/mem\_total)\*100;
17. mem\_total=mem\_total/(1024\*1024);
18. p->total=mem\_total;
19. p->used\_rate=mem\_used\_rate;
20. fclose(fd);     //关闭文件fd
21. **return** p ;
22. }

3.获取硬盘的使用情况

**[cpp]** [view plain](https://blog.csdn.net/zyy617532750/article/details/59145693) [copy](https://blog.csdn.net/zyy617532750/article/details/59145693)

1. **typedef** **struct** DEV\_MEM         //定义一个mem occupy的结构体
2. {
3. **double** total,used\_rate;
4. }DEV\_MEM;

主程序：

**[cpp]** [view plain](https://blog.csdn.net/zyy617532750/article/details/59145693) [copy](https://blog.csdn.net/zyy617532750/article/details/59145693)

1. DEV\_MEM \*get\_devmem()        // get hard disk meeeage
2. {
3. **FILE** \* fp;
4. **int** h=0;
5. **char** buffer[80],a[80],d[80],e[80],f[80],buf[256];
6. **double** c,b;
7. fp=popen("df","r");
8. fgets(buf,256,fp);
9. **double** dev\_total=0,dev\_used=0;
10. DEV\_MEM  \*dev=(DEV\_MEM \*)malloc(**sizeof**(DEV\_MEM));
11. **while**(6==fscanf(fp,"%s %lf %lf %s %s %s",a,&b,&c,d,e,f))
12. {
13. dev\_total+=b;
14. dev\_used+=c;
15. }
16. dev->total=dev\_total/1024/1024;;
17. dev->used\_rate=dev\_used/dev\_total\*100;
18. pclose(fp);
19. **return** dev;
20. }

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/zyy617532750/article/details/59145693