



C++程序设计

实验报告

专 业： 电子信息类

学 号： 20188068

班级序号： 180409

姓 名： 孔天欣

提交日期： 2018 年 12 月 日

成 绩：

东北大学秦皇岛分校计算机与通信工程学院



实验编号：5

实验名称：自定义数据类型

实验目的：

- (1) 掌握结构体类型变量的定义和使用；
- (2) 掌握结构体类型数组的概念和应用；
- (3) 了解链表的概念，初步学会对简单的链表进行操作。

实验内容：

设计一个结构体类型，描述进程的结构，如图 1 所示。然后定义一个结构体数组或链表存储如图 2 所示的 4 个进程。初始时进程的状态都是就绪状态即 $pState=1$ ，从优先级最高的进程开始执行，将执行的进程状态 $pState$ 由就绪状态改为执行状态即 $pState=2$ 。然后不断循环，每循环一次，就绪状态的进程优先级增加 1，其余不变；执行状态的进程优先级减 3 且其 $pCPU$ 减 1；当 $pCPU$ 为 0 时，该进程执行完毕，其进程状态修改为停止即 $pState=3$ ，其余不再变化。直到所有进程 $pCPU$ 都为 0，则循环结束。

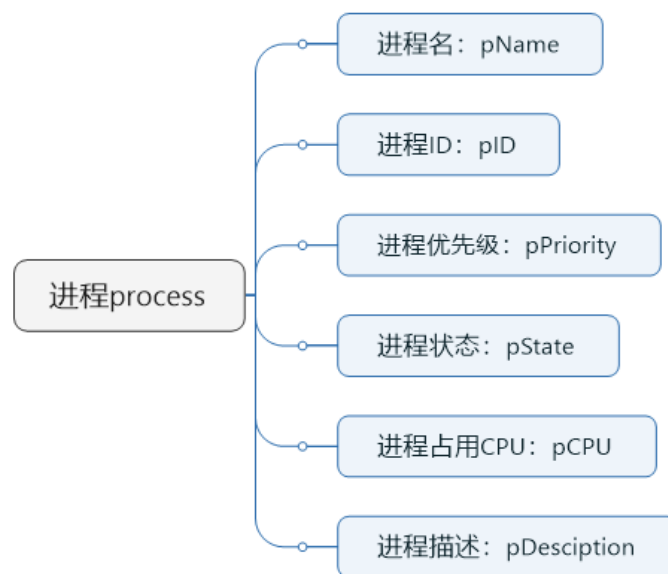


图 1. 进程的结构

进程按照优先级从大到小排序

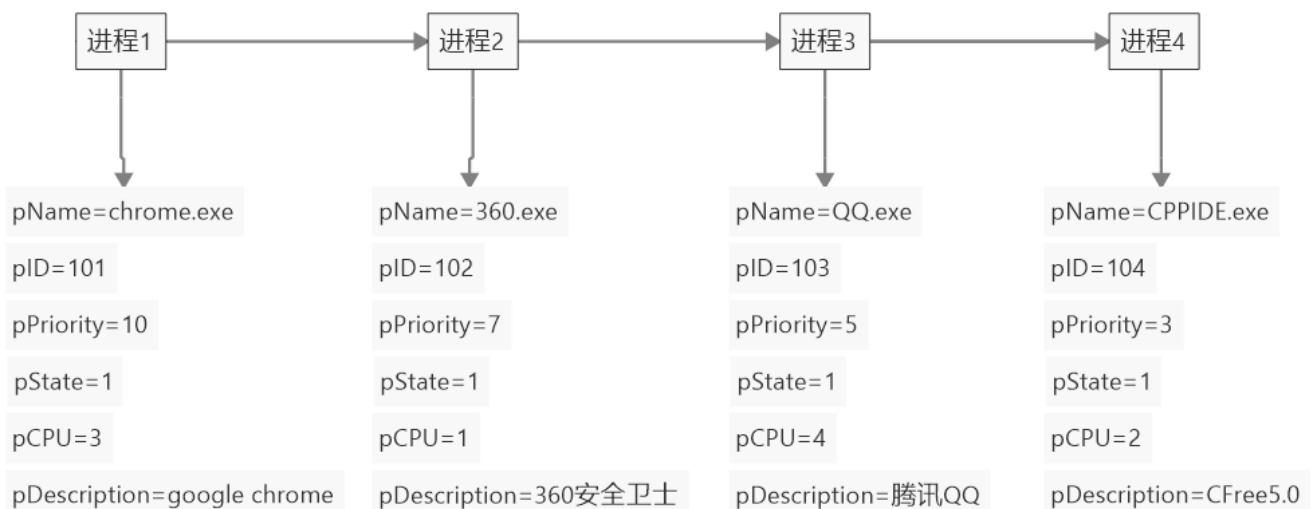


图 2. 4 个进程对象



实验结果:

```
#include<iostream>
#include<string>
#include"stdio.h"
using namespace std;

struct process
{
    string pName;
    int pID;
    int pPriority;
    int pState;
    int pCPU;
    string pDescription;
    struct process *next;
};

int main()
{
    int m, n, judge = 0;
    process proc[5], tmax = { "", 0, -1, 0, 0, "" };
    proc[0] = { "chrome.exe", 101, 10, 1, 3, "google chrome" };
    proc[1] = { "360.exe", 102, 7, 1, 1, "360安全卫士" };
    proc[2] = { "QQ.exe", 103, 5, 1, 4, "腾讯QQ" };
    proc[3] = { "CPPIDE.exe", 104, 3, 1, 2, "CFree5.0" };
    while (1)
    {
        tmax = { "", 0, -999, 0, 0, "" };
        for (m = 0; m <= 3; m++)
        {
            if (tmax.pPriority <= proc[m].pPriority)
            {
                tmax = proc[m];
            }
        }
    }
}
```



```
for (m = 0; m <= 3; m++)
{
    if (tmax.pName == proc[m].pName)
    {
        proc[m].pState = 2;
        break;
    }
}

while (proc[m].pCPU != 0)
{
    for (int n = 0; n <= 3; n++)
    {
        if (proc[n].pState == 1)
        {
            proc[n].pPriority++;
        }
    }

    proc[m].pPriority -= 3;
    proc[m].pCPU--;

    for (int i = 0; i <= 3; i++)
    {
        cout << "pName: " << proc[i].pName << " pID: " << proc[i].pID
            << " pPriority: " << proc[i].pPriority << " pState: " << proc[i].pState
            << " pCPU: " << proc[i].pCPU << " pDescription: "
            << proc[i].pDescription << " " << endl;
    }
    cout << endl;
}

proc[m].pState = 3;
for (int j = 0; j <= 3; j++)
{
    if (proc[j].pCPU == 0)
    {
        judge++;
    }
}

if (judge == 4)
{
    break;
}
else
{
    judge = 0;
}

cout << endl;
}
```



```
proc[m].pState = 3;
for (int j = 0; j <= 3; j++)
{
    if (proc[j].pCPU == 0)
    {
        judge++;
    }
}

if (judge == 4)
{
    break;
}

else
{
    judge = 0;
}

cout << endl;

cout << "kongtianxin 20188068";
```

```
pName: 360.exe pID: 102 pPriority: 7 pState: 3 pCPU: 0 pDescription: 360安全卫士
pName: QQ.exe pID: 103 pPriority: 3 pState: 2 pCPU: 2 pDescription: 腾讯QQ
pName: CPPIDE.exe pID: 104 pPriority: 9 pState: 1 pCPU: 2 pDescription: CFree5.0

pName: chrome.exe pID: 101 pPriority: 1 pState: 3 pCPU: 0 pDescription: google chrome
pName: 360.exe pID: 102 pPriority: 7 pState: 3 pCPU: 0 pDescription: 360安全卫士
pName: QQ.exe pID: 103 pPriority: 0 pState: 2 pCPU: 1 pDescription: 腾讯QQ
pName: CPPIDE.exe pID: 104 pPriority: 10 pState: 1 pCPU: 2 pDescription: CFree5.0

pName: chrome.exe pID: 101 pPriority: 1 pState: 3 pCPU: 0 pDescription: google chrome
pName: 360.exe pID: 102 pPriority: 7 pState: 3 pCPU: 0 pDescription: 360安全卫士
pName: QQ.exe pID: 103 pPriority: -3 pState: 2 pCPU: 0 pDescription: 腾讯QQ
pName: CPPIDE.exe pID: 104 pPriority: 11 pState: 1 pCPU: 2 pDescription: CFree5.0

pName: chrome.exe pID: 101 pPriority: 1 pState: 3 pCPU: 0 pDescription: google chrome
pName: 360.exe pID: 102 pPriority: 7 pState: 3 pCPU: 0 pDescription: 360安全卫士
pName: QQ.exe pID: 103 pPriority: -3 pState: 3 pCPU: 0 pDescription: 腾讯QQ
pName: CPPIDE.exe pID: 104 pPriority: 8 pState: 2 pCPU: 1 pDescription: CFree5.0

pName: chrome.exe pID: 101 pPriority: 1 pState: 3 pCPU: 0 pDescription: google chrome
pName: 360.exe pID: 102 pPriority: 7 pState: 3 pCPU: 0 pDescription: 360安全卫士
pName: QQ.exe pID: 103 pPriority: -3 pState: 3 pCPU: 0 pDescription: 腾讯QQ
pName: CPPIDE.exe pID: 104 pPriority: 5 pState: 2 pCPU: 0 pDescription: CFree5.0
```

kongtianxin 20188068

C:\Users\Dell\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe (进程 12000)已退出, 返回代码为: 0。
若要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...

出来源(S): 调试

oleApplication1.exe" (Win32): 已加载 "C:\Windows\SysWOW64\ntsvcr.dll"

心得体会:



实验编号: 6

实验名称: 类和对象

实验目的:

- (1) 掌握声明类的方法, 类和类的成员的概念以及定义对象的方法;
- (2) 初步掌握用类和对象编制基于对象的程序;
- (3) 学习检查和调试基于对象的程序;
- (4) 掌握多文件的程序的编译和运行的方法。

实验内容:

创建一个学生类, 成员变量有姓名, 学号和两门课 c++和高等数学的成绩, 成员函数有求最高分和平均分的函数: `maxCplus()`, `maxAMath()`, `avgCplus()`, `avgAMath()`。如图 1 所示。建立一个对象数组或链表, 存储 5 个学生的学号, c++成绩和高等数学成绩, 用指向对象的指针作函数参数, 在两个 `max` 函数中分别找出 5 个学生的该门课程的最高成绩者, 并输出其姓名、学号和相应课程的成绩。在两个 `avg` 函数中分别计算出该门课程的平均分并输出课程名和平均分。

要求: 创建一个工程(project), 将 `Student` 类的声明放到头文件 `student.h` 中, 将 `Student` 类的成员函数的定义放到源文件 `student.cpp` 中, 将 5 个学生对象的初始化以及最高成绩和平均成绩的计算输出写到源文件 `main.cpp` 中。如图 2 和 3 所示。从而学习多文件程序的编写。

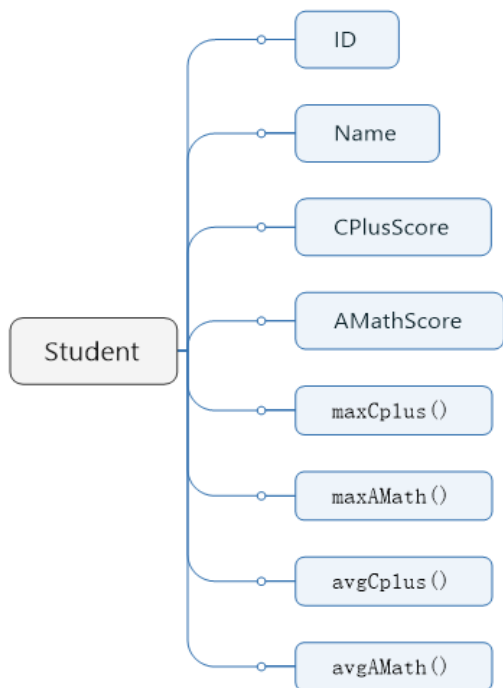


图 1

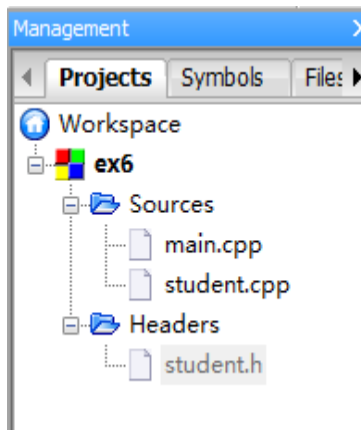


图 2

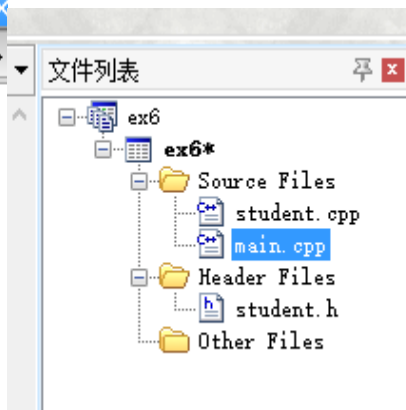


图 3

实验结果:

心得体会: