

编译系统原理第一章预习作业

计科一班 2013536 汤清云

1. C 程序优化:

代码具体如下:

```
#include<iostream>
#include<windows.h>
#include<stdlib.h>
#include<time.h>
using namespace std;
int main()
{
    int N=1000;
    //cin >> N;
    int* a = new int[N];
    long counter;
    float seconds;
    int step = 10;
    clock_t start, finish;
    start = clock();
    counter = 0;
    //初始化
    for (int i = 0; i < N; i++)
    {
        a[i] = 100;
    }
    /*loop1
    for (N = 0; N <= 1000; N += step) {
        // get time for size n
        start = clock();
        counter = 0;
        while (clock() - start < 10) {
            counter++;
            for (int i = 0; i < N; i++)
            {
                a[i] = a[i] * 2000;
                a[i] = a[i] / 10000;
            }
        }
        finish = clock();
        seconds = (finish - start) / float(CLOCKS_PER_SEC);
        cout << N << ' ' << counter << ' ' << seconds << ' ' << seconds /
```

```

counter << endl;
    if (N == 100) step = 100;
}
*/

/*loop2*/
int* b;
b = a;
for (N = 0; N <= 1000; N += step) {
    // get time for size n
    start = clock();
    counter = 0;
    while (clock() - start < 10) {
        counter++;
        for (int i = 0; i < N; i++)
        {
            *b = (*b) * 2000;
            *b = *b / 10000;
            b++;
        }
        b = a;
    }
    finish = clock();
    seconds = (finish - start) / float(CLOCKS_PER_SEC);
    cout << N << ' ' << counter << ' ' << seconds << ' ' << seconds /
counter << endl;
    if (N == 100) step = 100;
}

}

```

测试结果如下：

对于第一种程序，N 的取值、测试次数、总共运行时间/s、平均运行时间/s 如下：

```
选择 Microsoft Visual Studio 调试控制台
0 93235 0.01 1.07256e-07
10 72064 0.01 1.38766e-07
20 37598 0.01 2.65972e-07
30 40358 0.01 2.47782e-07
40 33458 0.01 2.98882e-07
50 25948 0.01 3.85386e-07
60 25823 0.01 3.87252e-07
70 20921 0.01 4.77989e-07
80 20462 0.01 4.88711e-07
90 17735 0.01 5.63857e-07
100 17224 0.01 5.80585e-07
200 8941 0.01 1.11844e-06
300 6263 0.01 1.59668e-06
400 4639 0.01 2.15564e-06
500 3742 0.01 2.67237e-06
600 2979 0.01 3.35683e-06
700 2416 0.01 4.13907e-06
800 2441 0.01 4.09668e-06
900 2153 0.01 4.64468e-06
1000 1980 0.01 5.05051e-06
E:\junior1\Compile\作业\Test1\Debug\Test1.exe (进程 20440) 已退出，代码为 0。
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。...
```

精确计时（单位毫秒）：

```
选择 Microsoft Visual Studio 调试控制台
0 2000 0.0058 2.9e-06
10 2000 0.1312 6.56e-05
20 2000 0.2139 0.00010695
30 2000 0.252 0.000126
40 2000 0.5904 0.0002952
50 2000 0.4679 0.00023395
60 2000 0.6446 0.0003223
70 2000 0.744 0.000372
80 2000 0.9718 0.0004859
90 2000 0.7881 0.00039405
100 2000 1.1538 0.0005769
200 2000 2.1619 0.00108095
300 2000 3.5695 0.00178475
400 2000 3.8663 0.00193315
500 2000 4.6345 0.00231725
600 2000 5.8405 0.00292025
700 2000 7.6167 0.00380835
800 2000 7.3355 0.00366775
900 2000 9.592 0.004796
1000 2000 8.9974 0.0044987
E:\junior1\Compile\作业\Test1\Debug\Test1.exe (进程 19904) 已退出，代码为 0。
```

对于第二种程序，N 的取值、测试次数、总共运行时间/s、平均运行时间/s 如下：

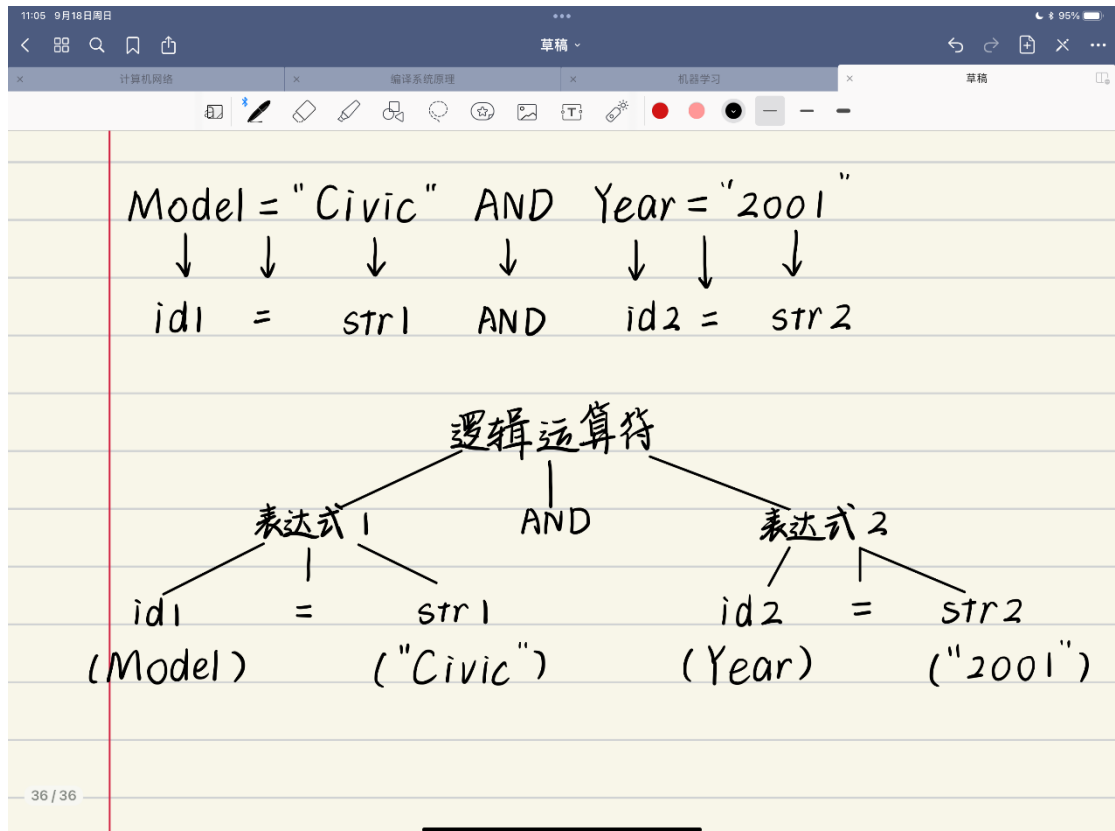
```
Microsoft Visual Studio 调试控制台
0 88491 0.01 1.13006e-07
10 75866 0.01 1.31811e-07
20 52542 0.01 1.90324e-07
30 41138 0.01 2.43084e-07
40 30837 0.01 3.24286e-07
50 28464 0.01 3.51321e-07
60 23845 0.01 4.19375e-07
70 22675 0.01 4.41014e-07
80 20184 0.01 4.95442e-07
90 17920 0.01 5.58036e-07
100 17920 0.01 5.58036e-07
200 8130 0.01 1.23001e-06
300 6468 0.01 1.54607e-06
400 4929 0.01 2.02881e-06
500 3553 0.01 2.81452e-06
600 3151 0.01 3.1736e-06
700 2556 0.01 3.91236e-06
800 1814 0.01 5.51268e-06
900 2384 0.01 4.19463e-06
1000 2032 0.01 4.92126e-06
```

精确测试（单位毫秒）：

```
Microsoft Visual Studio 调试控制台
0 2000 0.0063 3.15e-06
10 2000 0.1302 6.51e-05
20 2000 0.2834 0.0001417
30 2000 0.3529 0.00017645
40 2000 0.5022 0.0002511
50 2000 0.5397 0.00026985
60 2000 0.5167 0.00025835
70 2000 0.7589 0.00037945
80 2000 0.8817 0.00044085
90 2000 0.9828 0.0004914
100 2000 1.1026 0.0005513
200 2000 1.75 0.000875
300 2000 3.2016 0.0016008
400 2000 3.9783 0.00198915
500 2000 5.0717 0.00253585
600 2000 7.6195 0.00380975
700 2000 7.0205 0.00351025
800 2000 7.4743 0.00373715
900 2000 8.5675 0.00428375
1000 2000 8.2172 0.0041086
```

实践所得，第二种更快，但是测试结果并不稳定。

2. 分词、构造语法树：



3. 静态检查:

```
#include <stdio.h>

char firstChar1 ( /*@null@*/ char *s)
{
    return *s; #可能为空的指针，函数可能返回 null
}

int main()
{

    return 0;
}
```

```

E:\junior1\Compile\tests_work>splint test.c
Splint 3.0.1.6 --- 11 Feb 2002

Cannot find standard library: standard.lcd
Check LARCH_PATH environment variable.
test.c: (in function firstChar1)
test.c(5,12): Dereference of possibly null pointer s: *s
A possibly null pointer is dereferenced. Value is either the result of a
function which may return null (in which case, code should check it is not
null), or a global, parameter or structure field declared with the null
qualifier. (Use -nulldef to inhibit warning)
test.c(3,36): Storage s may become null

Finished checking --- 1 code warning
E:\junior1\Compile\tests_work>_

```

```
#include <stdio.h>
```

```
int *glob;
```

```
int *f(int **x)
```

```
{ int sa[2] = { 0, 1 };
```

```
int loc = 3;
```

```
glob = &loc;
```

```
*x = &sa[0];#三、可从参数 x 访问堆栈分配的存储*x 而存储*x 变成堆
栈
```

```
return &loc;# 一、当函数返回时，由外部引用指向堆栈引用。调用之
后，堆栈分配的存储被销毁，留下一个悬空引用；二、直接地址&loc 仅
隐式返回:&loc 与直接地址(& operator 的结果)传输不一致。
```

```
}
```

```
#四、函数返回引用已释放存储的全局 glob 在传递控制时，全局变量不
满足其注释。
```

```
#五、声明被导出，但不在此模块之外使用。声明可以使用静态限定符。
```

```
int main()
```

```
{
```

```
return 0;
```

```
}
```

```

E:\junior1\Compile\tests_work>splint test.c
Splint 3.0.1.6 --- 11 Feb 2002

Cannot find standard library: standard.lcd
Check LARCH_PATH environment variable.
test.c: (in function f)
test.c(9,11): Stack-allocated storage &loc reachable from return value: &loc
A stack reference is pointed to by an external reference when the function
returns. The stack-allocated storage is destroyed after the call, leaving a
dangling reference. (Use -stackref to inhibit warning)
test.c(9,11): Immediate address &loc returned as implicitly only: &loc
An immediate address (result of & operator) is transferred inconsistently.
(Use -immediatetrans to inhibit warning)
test.c(9,16): Stack-allocated storage *x reachable from parameter x
test.c(8,4): Storage *x becomes stack
test.c(9,16): Function returns with global glob referencing released storage
A global variable does not satisfy its annotations when control is
transferred. (Use -globstate to inhibit warning)
test.c(9,11): Storage glob is released
test.c(3,6): Variable exported but not used outside test: glob
A declaration is exported, but not used outside this module. Declaration can
use static qualifier. (Use -exportlocal to inhibit warning)

Finished checking --- 5 code warnings

```

```

#include <stdio.h>
void h(void)
{ unsigned int i;
  if (i >= 0) #无符号值 i 与 0 不可以比较, 且 i 未定义
    printf(">=0\n");
  else printf("<0");#不可识别的标识符
}
int main()
{
  return 0;
}

```

```

E:\junior1\Compile\tests_work>splint test.c
Splint 3.0.1.6 --- 11 Feb 2002

Cannot find standard library: standard.lcd
Check LARCH_PATH environment variable.
test.c: (in function h)
test.c(4,9): Comparison of unsigned value involving zero: i >= 0
An unsigned value is used in a comparison with zero in a way that is either a
bug or confusing. (Use -unsignedcompare to inhibit warning)
test.c(4,9): Variable i used before definition
An rvalue is used that may not be initialized to a value on some execution
path. (Use -usedef to inhibit warning)
test.c(5,7): Unrecognized identifier: printf
Identifier used in code has not been declared. (Use -unrecog to inhibit
warning)

Finished checking --- 3 code warnings

```

4. 标识符列表的递归定义:

基本规则: 不以数字开头的只包含字母、下划线和数字的变量名是标识符; 单个标识符即是一个标识符列表。

递归定义：标志符列表+英文逗号+标志符列表的组合结果也是标志符列表