

Com 4521与GPU的并行计算：实验01

2019年春季学期

Mozhgan Kabiri Chimeh

博士

实验室助理: Robert Chisholm, John Charlton,

谢菲尔德大学计算机科学系

学习成果

- 了解如何在Visual Studio中编译和执行简单的C程序
- 了解错误输出以及如何解决它们
- 了解如何链接外部C库
- 理解文件和字符串操作
- 了解如何手动分配和释放内存
- 了解如何从二进制文件中读取
- 了解如何在C中实现链表数据结构。

修订

- 07/02/2018 - 练习2.5阐明了线性发电机的形式。

实验室注册

完成练习后，每个学生必须完成实验室注册。完成实验后，您应该完成此操作，包括查看解决方案。您不应该在实验课上完成此课程，但您应该在教学周结束时完成。

实验室注册链接: [埃克斯://gog. gl/0r73gd](https://gog.gl/0r73gd)

练习1

通过右键单击解决方案并选择Add-> New Project ...将新的Windows 32控制台项目添加到名为'Lab01_Exercise01'的解决方案中。选择它作为我们要执行的项目，右键单击项目并选择'设置为启动项目'。添加一个名为'exercise1.c'的新源文件，并从我们的hello world示例中复制内容。您可以删除hello world print命令。复制提供的'random.h'文件并将其放在项目源目录中。通过右键单击解决方案资源管理器中的项目并选择Add-> Existing item, 将random.h头文件添加到项目中。

我们将创建一个程序来创建规范化随机整数列表。

- 1.1 创建名为NUM_VALUES的预处理器定义，并为其指定值250。
- 1.2 使用预处理器定义声明一个名为"value"的全局签名32位整数数组，以定义数组大小。
- 1.3 在main函数中定义一个名为'sum'的本地无符号32位整数变量，该变量只能保存正值并将其初始化为0。

- 1.4 使用数据类型在主函数中定义局部变量*i*，该数据类型可以保存0-255范围内的值，将其初始化为0。
注意：如果声明变量并在不同的行上设置它们的值（即定义它们），那么您需要确保声明和定义都出现在任何表达式之前（例如sum = 0;）。
- 1.5 我们需要调用一个在头文件random.h中声明的函数，包含头文件并在main函数中调用init_random。
- 1.6 在0和NUM_VALUES范围内写一个简单的for循环（使用整数*i*作为计数器）。在循环内调用函数random_ushort并将值保存在索引*i*的values数组中。在循环中，为控制台创建一个print语句，该语句在一行中输出*i*的值和存储在数组中的值。我们可以用它来调试输出。
- 1.7 random_ushort函数包含从int到unsigned short的隐式转换。修改它以使它使用显式强制转换。这不会改变计划，但这是一种很好的做法。
- 1.8 通过注释掉调试语句并将值汇总到变量“sum”来修改循环。返回循环后输出和值。总和是多少？它应该是4125024. 使用适当的数据类型添加新的局部变量‘average’。计算并存储随机数的平均值。
- 1.9 通过减去平均值来标准化随机数。计算最小值和最大值。三元运算符

(条件表达式) ? 表达式 _ t: 表达式 _ f;

如果条件为true或者expression_f是条件为false，则可以用于返回expression_t。例如int x = (a == 0) ? 123 : 567;在计算最小值和最大值时使用此简写概念。打印平均值，最小值和最大值以及总和。您应该获得以下值。

总和= 4125024
平均值= 16500最
小值= -16247最
大值= 16221

练习2

我们现在将通过实施更好的随机函数来扩展上一个练习。将新的Windows 32控制台项目添加到名为“Lab01_Exercise02”的解决方案中。确保文件夹位置是解决方案的根目录（而不是以前的项目之一）。将新项目设置为启动项目，因此在运行程序时，它将不会运行上一个项目。复制上一练习中的源文件和头文件（将exercisel.c重命名为exercise2.c）并将它们移动到新项目源文件夹中。通过选择Add-> Existing Items将它们添加到项目中。

现有rand函数的问题在于，尽管返回32位整数，它仅返回0-32767范围内的值（有符号短整数的正范围）。这是因为Microsoft在首次实现时（以及当16位整数更常见时）保留了与使用该函数的代码的向后兼容性。这是msvc运行时的“特性”。你会发现在Linux中rand返回一个完整的32位数字。

2.1 让我们首先将随机函数定义分离为标题和单独的源模块。在项目中创建一个名为random.c的新文件。将init_random和random_ushort函数定义移动到新的源模块中。将stdlib.h的包含移动到新的源模块，并在random.c中包含random.h。random.h文件现在应该只包含种子。构建应用程序。编译器应该发出警告，指出这两个函数是未定义的。

2.2 通过添加适当的函数声明来修改random.h。如果不包含extern关键字，则它将由编译器隐式定义（因为默认情况下所有全局变量都是extern）。包含它是一种很好的做法。该项目现在应该没有错误地构建。

2.3 我们现在要实现简单的线性同余生成器（[维基链接](#)）。这通过简单的乘法和加法来推进种子。使用预处理器宏在random.h中定义参数RANDOM_A和RANDOM_C。将它们的值分别设置为1103515245和12345。

2.4 在random.c中创建一个名为'rseed'的32位无符号全局整数变量。修改init_random将rseed的值设置为RAND_SEED。

2.5 为函数创建函数声明（在random.h中）和定义（在random.c中）
random_uint返回unsigned int。使用等式实现线性生成器

$$x_{(n+1)} = A * x_{(n)} + C$$

其中x可以使用变量'rseed'（初始值 $x_0 = \text{RAND_SEED}$ ，根据练习2.4），A和C是你的参数。

2.6 在exercise2.c中调用random_uint替换对random_ushort的调用。我们的变量总和现在太小而无法保持总和值。将其修改为64位无符号整数，并确保将其正确打印到控制台。将values数组的类型也修改为64位有符号整数。您还需要通过添加显式强制转换来修改平均值，最小值和最大值的计算。确保您的printf格式对于数据类型是正确的。总和，平均值，最小值和最大值是多少？他们应该是：

总和= 524529029501

平均= 2098116118最小=

-2093214053最大=

2107382890

练习3

我们现在将通过实现浮点随机函数来扩展上一个练习。将新的Windows 32控制台项目添加到名为“Lab01_Exercise03”的解决方案中。确保文件夹位置是解决方案的根目录（而不是以前的项目之一）。将新项目设置为启动项目。复制上一练习中的源文件和头文件（将exercise3.c重命名为exercise4.c）并将它们移动到新项目源文件夹中。通过选择Add-> Existing Items将它们添加到项目中。

3.1 添加一个新函数定义和声明（random_float），返回一个随机浮点数。这应该是random_uint函数的值。修改示例，以便计算总和，平均值，最小值和最大值的浮点值。确保使用0位小数打印值。总和，平均值，最小值和最大值是多少？他们应该是：

总和= 524529139712

平均= 2098116608最小=

-2093214592最大=

2170532608

练习4

您将创建一个从命令行获取输入的计算器。

通过右键单击解决方案并选择Add-> New Project ...将新的Windows 32控制台项目添加到名为'Lab01_Exercise04'的解决方案中。选择它作为我们想要执行的项目，右键单击项目并选择set它作为启动项目。制作文件exercise4.c（为您提供）的副本，并将其放在项目源目录中。通过选择Add-> Existing Items将文件添加到项目中。

源文件包含一个简单的命令行计算器的基本结构，它将理解以下基本命令“add N”，“sub N”，“mul N”，“div N”和“exit”，其中N是一个浮点值。

- 4.1 通过向缓冲区顺序添加字符来完成while循环。
- 4.2 实施检查以确保不写入超出缓冲区限制的末尾。写入传递数组的结尾称为溢出。检测到潜在溢出时，使用fprintf向stderr写入错误消息，然后调用exit (1) 强制程序提前终止。
- 4.3 确保一旦while循环退出，缓冲区就会被字符串终止字符正确终止。
- 4.4 使用strcmp函数测试该行是否为“exit”。如果是，则readLine应返回0，否则应返回1。测试程序。当用户进入“退出”时它应该退出，否则它应该打印“未知命令”。
- 4.5 修改main函数中的while循环。检查该行是否包含三个字符后跟一个空格。您可以使用'ctype.h'中的isalpha函数来检查字符是否为字母。如果该行不符合此条件，则向stderr输出错误“不正确的命令格式”并使用continue再次开始循环。
- 4.6 假设满足4.5的条件，则使用sscanf将缓冲区中的3字符命令和浮点值分别提取到命令和a。

注意：您需要将in_value传递给前缀为&运算符的sscanf。例如
sscanf (... , &in_value) 。

- 4.7 修改条件false以检查命令以查看它是否等于“add”
- 4.8 为sub, mul和div创建else if条件。测试你的程序。
- 4.9 使用strncmp添加其他条件以测试命令的前两个字母。如果是“ad”，则输出“你的意思是添加吗？”完整的情况为“su”，“mu”，“di”。测试你的程序。

练习5

我们现在将通过修改计算器来扩展上一个练习，以便它可以从文件中读取命令。将新的Windows 32控制台项目添加到名为“Lab01_Exercise05”的解决方案中。确保文件夹位置是解决方案的根目录（而不是以前的项目之一）。将新项目设置为启动项目。从新项目源文件夹中的上一个练习中复制源文件。将源文件从exercise4.c重命名为exercise5.c。通过选择Add-将源文件添加到项目中
>现有项目。

- 5.1 修改示例，以便它可以读取提供的'commands.calc'文件。您需要实现以下内容；
 - 5.1.1 以只读模式打开和关闭文件

- 5.1.2 修改readLine函数，使其读取文件而不是控制台。您应该检查文件结束字符（EOF），如果找到则返回0。注意：此行为要求任何.calc文件在文件末尾都有一个空行。
- 5.1.3 修改主要功能。不正确的命令或拼写错误命令应导致控制台错误并立即退出。while循环应该是静默的（没有控制台输出），只有最终的总和应该输出到控制台。正确答案是99.0000。

练习06

本练习的目的是修改一些现有代码以使用指针。我们为您提供了一些示例代码。示例代码将读入二进制文件，其中包含4条关于学生的信息记录。这些信息包括他们的姓，姓和平均模块标记。已定义一个结构来保存学生数据，此结构的格式与创建示例中使用的二进制文件的程序中使用的结构相匹配。

- 1.1 编译并执行程序。它应该打印出4名学生的信息。
- 1.2 print_student函数效率低下。它需要传递一个结构（按值），这会导致所有数据被复制。对此进行修改，以便将结构作为参考传递。您需要更新print_student函数声明和定义。
- 1.3 main函数使用静态定义的数组来保存学生数据。修改此代码，以便学生是指向学生结构的指针，然后手动分配足够的内存来读取学生记录。不要忘记在程序结束时释放数据。

练习07

将您之前的代码从第一个练习复制到名为“Lab01_Exercise07”的新项目中。

学生结构使用静态定义的固定长度char数组来保存forename和surname。这是可以的，但是当我们处理大型记录时，可能会浪费，因为很多char数组都是空的。student2.bin文件与第一个练习中使用的文件的不同之处在于它使用动态长度char数组来保存字符串。forename和surname都以下列格式写入二进制文件：

```
unsigned int n, char [0], char [1], char [2], ..., char [n];例如  
5, 'J', 'o', 'h', 'n', '\0'
```

修改结构定义，使forename和surname成为char的指针。现在更新代码以读取学生数据。您将需要使用fread来读取forename的长度（即n）。提示：为forename（长度为n）分配内存，然后查找forename等。不要忘记更新代码以确保释放已分配的内存。

练习08

将上一个练习中的上一个代码复制到名为“Lab01_Exercise08”的新项目中。

之前的练习都假设我们知道有多少学生记录存储在二进制数据文件中。对于下一个练习，我们将更新我们的程序以读取，存储和显示

任意数量的记录。为此，我们将使用链表数据结构。linked_list.h头文件包含通用链表的非常基本的实现。头文件包含一个结构llitems，它定义了一个指向列表中上一个和下一个项的指针。

1.1 链表的实现是不完整的。完成功能

add_to_linked_list () 通过实现以下内容；

- 1.1.1 检查ll_end项目实际上是列表的末尾（下一条记录应该是空值）。如果它不是结束那么函数应该返回NULL。
- 1.1.2 将项添加到链表的末尾，更新链表的旧端以反映添加。
- 1.1.3 返回指向链表新端的指针。

1.2 为了使用print_items函数，必须将函数指针print_callback设置为具有以下声明的函数；

打印功能_暴力（行为）；

你已经有了一个可以使用的函数print_student，但是这个函数接受了一个指向学生结构的const指针。使用显式强制转换为print_callback函数指针分配print_student函数。在这里你必须小心使用括号。

1.3 通过创建学生记录的链接列表，更新您的代码以在students2.bin中阅读。您将需要一个指针来标记链表的开头和结尾。要测试您的流是否在文件末尾（即它已读取最后一条记录），您应该检查fread的返回值（如果小于请求的项目数，则表示文件结束）。您应该使用create_linked_list和add_to_linked_list函数。您可以使用free_linked_list函数释放链接列表，但要小心，因为这不会释放链接列表指向的记录。