密级: 公 开

信息工程大学

操作系统实验报告

Linux初识

单 位： 网络空间安全学院

队 别： 十一大队一队

学 号： 4042017030

姓 名： 王珺熠

学科专业： 网络工程

日 期： 2019 年 9 月 6 日

说 明

一、本实验的目的是帮助学员熟悉Linux下基本命令行使用、gcc编译器的使用等操作。

二、 要求学员使用C语言进行代码实现下面的题目。

**数据说明：**

总共包括3个数据文件：customer.txt, lineitem.txt和orders.txt。分别行数为100，1000和4000，列和列之间使用|来分隔。所有文件均为文本文件。

每一个顾客在customer表中有一个唯一的顾客编号c\_custkey，和对应的某个市场部门c\_mkgsegment。每个顾客可能会有任意多个历史订单，每个订单在orders表中有一行，它有一个唯一的订单号o\_orderkey和下单顾客编号o\_custkey和一个下单时间c\_orderdate。每个订单又由多件商品组成，每件商品在lineitem中有一行，记录了它所属的订单号l\_orderkey、价格l\_extendedprice及发货时间l\_shipdate。

目标是找出每个订单延迟发货的商品的总销售额，以供销售方有针对性地分析优化。如下图所示，我们将该问题抽象为以下的一个计算问题：每张表看作一个multi-set，对customer、orders和lineitem三张表建立笛卡尔集合，得到一个mnl行的一个集合（其中m,n,l分别是三张表的行数），然后对该集合按以下条件做过滤：

c\_mktsegment = ？ and c\_custkey=o\_custkey and o\_orderkey=l\_orderkey and o\_orderdate < ? and l\_shipdate > ? ：其中？是动态变参

对满足过滤条件的结果，按照l\_orderkey列的值进行分组，并对l\_orderkey相同的记录的l\_extendedprice做求和，最后结果按照l\_extendedprice求和值排序，并返回topn的结果，其中topn值也是动态变参。

该计算过程，也可以使用以下SQL来描述：

select

l\_orderkey,

o\_orderdate,

sum(l\_extendedprice) as revenue

from

customer,

orders,

lineitem

where

c\_mktsegment = ？

and l\_orderkey = o\_orderkey

and c\_custkey = o\_custkey

and o\_orderdate < ？

and l\_shipdate > ？

group by

l\_orderkey,

o\_orderdate

order by

revenue desc

LIMIT ？

**提交要求：**

需要提供5个文件（夹）

Readme.txt：编译和运行方式 使用ubuntu自带的c语言编译器(gcc)

Compile.sh：编译脚本

Run.sh: 执行脚本

Sourcecode 文件夹：包括所有source code和配置文件

可执行文件：可执行文件

**程序执行方式：**

选手需要提前将3个数据文件放到run.sh 相同目录，提交的程序将会以如下格式的命令运行、进行评测。

./run.sh customer.txt orders.txt lineitem.txt n n4个参数

第四个参数表示总共计算的次数

第5~8个参数为第一次计算时的4个参数

第9~12个参数为第二次计算时的4个参数

…

第1+4n~4+4\*n 个参数为第n次计算时的4个参数

四个参数分别对应以下的条件值：

c\_mktsegment = ？

o\_orderdate < ？

l\_shipdate > ？

LIMIT ？

执行示例：

示例1：

./run.sh customer.txt orders.txt lineitem.txt 1 BUILDING 1995-03-29 1995-03-27 5

示例2：

./run.sh customer.txt orders.txt lineitem.txt 3 BUILDING 1995-03-29 1995-03-27 5 BUILDING 1995-02-29 1995-04-27 10 BUILDING 1995-03-28 1995-04-27 2

程序输出：所有结果按行输出，多列结果使用|分割

输出示例：

示例1：

l\_orderkey|o\_orderdate|revenue

249739810|1995-02-28|513890.13

180639074|1995-03-20|502044.66

202071367|1995-03-12|499760.62

31617348|1995-03-27|498949.38

461984355|1995-03-13|491959.42

示例2：

l\_orderkey|o\_orderdate|revenue

249739810|1995-02-28|513890.13

180639074|1995-03-20|502044.66

202071367|1995-03-12|499760.62

31617348|1995-03-27|498949.38

461984355|1995-03-13|491959.42

l\_orderkey|o\_orderdate|revenue

594317283|1995-02-11|457168.01

328093060|1995-02-03|453863.74

322499142|1995-02-25|427615.4

319693698|1995-02-05|425802.97

323689092|1995-02-23|416615.42

128919045|1995-02-28|414996.73

540656743|1995-02-23|411107.69

311469825|1995-02-21|402946.11

164796770|1995-02-27|401863.1

131043138|1995-02-20|399884.41

l\_orderkey|o\_orderdate|revenue

31617348|1995-03-27|498949.38

461984355|1995-03-13|491959.42

|  |  |  |
| --- | --- | --- |
| 一、基本情况 | | |
| 实验环境 | | 说明：使用的笔记本型号，虚拟机配置，Linux版本 |
| 二、实验过程  （一）对目标功能在Windows系统下的实现  **1.1设计思路**  经过对实验要求的分析，我觉得可以将程序的实现分成三部分进行实现：  1、对三个txt文件进行输入到结构体：customer、orders、lineitem当中。  2、对三个结构体进行级联生成新的结构体：select\_result,并对 extendedprice求和值排序。  3、根据题设要求进行三个参数：limit、order date、ship date的输入，并对筛选结果进行  **1.2程序实现**  1.2.1 读入txt内容  利用fopen函数将txt中的内容输入到预先设定的结构体中，针对txt中的无用项“|”，设定一个char型的c用于存取，对结构体内的值不产生影响。  1.2.2 判别、级联  在满足筛选条件如下时：  “cus[i].c\_custkey==ord[j].o\_custkey&&ord[j].o\_orderkey==item[k].l\_orderkey&&(strcmp(ord[j].o\_orderdate,order\_date)<0)&&(strcmp(item[k].l\_shipdate,ship\_date)>0)”  用select函数对result表进行赋值，根据要求进行冒泡排序操作并求和。  1.2.3实现输入与输出  用scanf函数进行limit、order date、ship date的输入，并将该三个参数的值在select函数中进行调用，进行输出时，当limit超出result表中有效组数时，输出会出现问题；加入一个判别条件（l.orderkey<0）（正常来说有意义的值应该大于零），对超出部分输出null。  **1.3源码（附于windows文件夹中）**  **1.4运行结果**  Txt表导入成功：  test • Microsoft Visual Studio  FCR\ITCRE  17 AUTOMOBILE  18 BUILDING  19 HOUSEHOLD  FURNITURE  1 WACHI.VERY  02 WACHINERY  03 HOUSEHOLD  04 WACHINERY  25 FURNITURE  AUTOMOBILE  BUILDING  FURNITURE  FURNITURE  30 BUILDING  31  HOUSEHOLD  32 BUILDING  33 AUTOMOBILE  34 HOUSEHOLD  35 HOUSEHOLD  36 BUILDING  37 FURNITURE  38 HOUSEHOLD  39 AUTOMOBILE  10 BUILDING  n  HOUSEHOLD  12 BUILDING  13 WACHINERY  M AUTOMOBILE  (Ctrl+Q)  int i -O,  orders * a=NÜLL:  a (orders * (orders)) ,  100 %  cpp(203): varning C4996: ' scanf' : This function or variable nay be Lmsafe.  files (x86) \nicrosoft visual studio Il. : "scanf"  scanf' : This function or variable nay be unsafe.  I cpp(205): varninz C499€:  c: iprogran files (x86) inicrosoft visual studio IL. : "scant"  I > test. vcxproj exe  04  Consi der  Consi der  using scanf_s instead. To  using scanf_s instead. TO  20:08  2019/9/8  输出结果：  test • Microsoft Visual Studio  0-0  Windows  experiment.cpp -u  SQL(Q)  Debug • Win32  EDU)  04  (Ctrl+Q)  select(customer * CuS, orders * ord, lineitem item, char * order date, char * ship_date)  x  Sl test  > experiment. cpp  result2[rnl. l_orderkey - resultl[i). l_orderkey;  strcpy (result2 Cm]. o_orderdate, resultl o_orderdate)  result2[rnl. l_extendedprice resultl[i). l_extendedprice,  input Limit:5  Input Order Date:  Input Ship Date:  I _orderkey o_orderdate revenue  450  43  386  997  928  1995-03-05  1995-03-25  1995-01-25  1995-03-20  166262. 09  308542. 80  109979. 70  99825. 59  363200. 54  20:03  2019/9/8  排序后结果：  test • Microsoft Visual Studio  0-0  Windows  experiment.cpp -u  SQL(Q)  EDU)  Debug • Win32  (Ctrl+Q)  select(customer * CuS, orders Ord, lineitem item, char * order date, char * ship_date)  x  Sl test  > experi  input Limit  Input Order Date: 1995—03—27  Input Ship Date: 1995—03—29  _ orderkey o_orderdate  928  775  643  450  450  1995-03-02  1995-03-18  1995-03-25  1995-03-05  1995-03-05  revenue  363200. 54  346899. oo  308542. 80  166262. 09  166262. 09  cpp(203): varning C4996: ' scanf' : This function or variable nay be Lmsafe.  files (x86) \nicrosoft visual studio Il. : "scanf"  scanf' : This function or variable nay be unsafe.  I cpp(205): varninz C499€:  files (x86) inicrosoft visual studio Il. : "scant"  I > test. vcxproj exe  04  Consider using scanf_s instead. To A  Consider using scanf_s instead. TO  20:03  2019/9/8  （二）对目标功能在ubuntu下的实现  2.1  在ubuntu实现于windows下的区别在于shell脚本下执行程序，需要进行传参，经过查阅教程，发现传参使用argc、argv[]进行实现  2.2脚本：  Run.sh: ./experiment.o $@  $@含义：所有参数列表。如"$@"用「"」括起来的情况、以"$1" "$2" … "$n" 的形式输出所有参数。  Compile.sh: gcc experiment.c -o experimen  编译experiment.c文件并生成对应的.o文件  2.3问题及解决  问题：在使用vim compile.sh创建编译脚本并在编译脚本内写下“gcc experiment.c -o experiment”之后，在ubuntu的terminal里输入。“./compile.sh”尝试执行脚本报错：bash: ./compile.sh: Permission denied提示脚本权限不够。  Activities Terminal  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  sun 03:13  test  •a  Videos  D  Home  Recent  Home  Desktop  Documents  Downloads  Music  Pictures  videos  Trash  Other Locations  compile.sh  customer.  Examples  lineitem.txt  •char  •char  •char  orders.txt  readme. txt  run.sh  joey@ubuntu: —  File Edit View Search Terminal Help  vim compile. sh  . /comptle. sh  bash: . /comptte.sh: permission denied  chmod 777 compile. sh  . "compile .sh  comptte.c: No such file or directory  gcc:  fatal error:  no input files  gcc:  compilation terminated.  vim compile. sh  src.c: In function :  warntng: format • expects argument Of type •char • '  fscanf(fp, "XdXc%s" , Ra[t]. c_custkey , 8b . c_mkgsegment);  but argument 5 has type  but argument 7 has type  src.c: In function 'read_orders_txt':  warntng: format • expects argument  Of  fscanf(fp, , Ra[tJ. o_orderkey , 8b , Rd , Rc ,  src.c: In function :  warntng: format • expects argument  Of  type  type  •char  .o_orderdate);  •char  but argument 7 has type  fscanf(fp, "XdxcXlfXc%s" , Rl[t] . l_orderkey . l_extendedprtce ,8b, t , l_shtpdate) ;  解决办法：  对脚本进行777赋权：777代表，user,group ,others ,都有读写和可执行权限。输入命令行：chmod 777 compile.sh  2.4运行结果Activities Terminal  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  sun 03:12  test  D  Home  Recent  Home  Desktop  Documents  Downloads  Music  Pictures  videos  Trash  Other Locations  •a  Videos  compile.sh  customer.  Examples  lineitem.txt  orders.txt  readme. txt  run.sh  Joey@ubuntu: -  File Edit View Search Terminal Help  ./run.sh customer. txt orders. txt txt 3 BUILDING 1995-03-29 1995-03-27 5 BUILDING 1995  -62-29 1995-04-27 16 BUILDING 1995-03-28 1995-04-27  l_orderkey I o_orderdate I revenue  450  359  577  null  nut t  11995-63-05 1166262.89  11994-12-19 158329.62  11994-12-19 141811.50  I null  I null  I null  I_orderkey I o_orderdate I revenue  nutt  null  null  nutt  null  nut t  null  null  nutt  null  I nut t  I null  I null  I null  line I:  I nut I  Inuu  Inuu  I null  Inuu  I nut I  Inuu  Inuu  I null  Inuu  3245 Segmentation fault  (core dumped) . /src.o  ./run.sh customer. txt orders. txt ltnettem. txt 1 BUILDING 1995-03-29 1995-03-27 5  Corder key I O_order date I revenue  450  359  577  null  null  11995-03-es 1166262.09  11994-12-19 158329.62  11994-12-19 141811.50  I null  Inuu  Inuu | | |
|  | | |
| 三、总结部分  描述通过实验课学到的知识和建议。 | | |