1 密级: 公 开

XX大学

操作系统实验报告

Linux初识

单 位： 三院

队 别： 八大队四队

学 号： 4062017011

姓 名： 黄庆华

学科专业： 信息安全

日 期： 2020 年 05 月 31 日

说 明

一、本实验目的为熟悉linux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m\*n\*l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

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：**

每名学生需要提前将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 | 电脑RedmiBook14 i5，虚拟机vmware，Linux版本deepin15.7 |
| 二、实验过程  1、搭建实验环境  安装虚拟机，我安装的是之前电脑中留存的deepin15.7，基本操作和Ubuntu没有什么区别，然后用命令sudo apt install vim/gcc在线安装vim和gcc  2、编写实验代码  其他部分使用教员给的代码，选择订单部分我打算使用条件函数进行选择，第一步，选出市场部门和用户匹配，第二步，匹配顾客编号和下单的顾客编号，第三步，匹配订单号和下单的订单号，都匹配成功后，将订单号、下单时间和销售额的数据打印出来，使用|隔开    3、执行代码文件  在终端中执行compile.sh编译代码src.c生成可执行文件  然后执行run.sh文件，输入四个参数，等待结果  示例一结果：    示例二结果： | |
| 三、总结部分  通过这次实验，我初步接触Linux系统，和Windows操作系统不一样的是Linux主要靠代码进行操作而Windows是靠UI界面来操作，所以一开始有点不适应。实验的过程还是有一些艰难，一大难关是C语言，要重新学习C语言的指针和结构。 | |