

USIM 卡文件系统中动态存储管理的研究

马海峰,李代平,尹伟,欧阳小星,李春丰

(广东工业大学 计算机学院,广东 广州 510006)

摘要:为了对智能卡内部有限的存储空间进行有效的分配和管理以适应实际的需求,对 USIM 卡文件系统中动态存储管理的研究是至关重要的。文中针对 NOR Flash 具有擦写次数限制、先擦后写的特性和 USIM 卡中多应用存储空间的分配特点,运用动静结合的理论,设计一种混合存储方式组织数据存储,并给出了存储空间的动态管理方法和具体算法描述。测试结果证实该方法适合 3G USIM 卡多应用的需求,保证智能卡文件系统的可靠性,提高 Flash 的利用率,最大化地延长 Flash 的使用寿命,有效提高数据的检索速度。

关键词:USIM 卡;动态存储管理;垃圾回收;均衡磨损

中图分类号:TP36

文献标识码:A

文章编号:1673-629X(2010)08-0071-04

USIM Card File System of Dynamic Memory Management

MA Hai-feng, LI Dai-ping, YIN Wei, OUYANG Xiao-xing, LI Chun-feng

(College of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: In order for smart cards within the limited storage space for effective allocation and management to meet the practical needs of the USIM card file system, research of dynamic storage management is essential. In this paper, NOR Flash has a write endurance limit, written after the first cleaning features and USIM card storage space allocation for multi-application features, the use of static and dynamic combination of theory, design a hybrid storage organizing data storage, and gives a dynamic storage management methods and specific description of the algorithm. Test results confirmed that the method is suitable for 3G USIM card multi-application requirements, to ensure the reliability of the smart card file system to improve the utilization of Flash, maximizing the life of the extension of Flash, effectively improve the speed of data retrieval.

Key words: USIM card; dynamic memory management; garbage collection; wear-leveling

0 引言

随着 3G 时代的到来,3G 产业得到了迅猛发展。对于中国联通来说,WCDMA 是目前国际上最为成熟的 3G 标准,这对中国联通的移动智能卡市场带来的变化也最大。

智能卡的应用日趋广泛,一张卡上集成多个应用的多应用智能卡^[1]成了研究的热门。基于 WCDMA 网络的多应用 USIM 卡,在使用过程中可以动态地下载和删除程序,这就需要 COS 通过 USIM 文件系统的管理和组织来实现对存储空间分配和回收的动态管理。这样能很好利用智能卡上有限的存储空间,减少了在查找数据及更新数据上所耗费的时间。

1 NOR Flash 存储器的特点

基于 Flash 的存储器是智能卡 COS 系统最佳的存储选择。目前,嵌入式系统中使用的 Flash 主要分为 NOR 和 NAND^[2,3]两种类型。其中 NOR Flash^[2]的主要特点是:

(1) NOR Flash 是非易失性存储设备,访问速度快,并在芯片内执行。即可以直接在 Flash 闪存内运行应用程序,不需要把代码读到系统 RAM 中。

(2) NOR Flash 带有 SRAM 接口,有足够的地址引脚来寻址,可以实现完全随机的字节读取。

(3) NOR Flash 具有独特的读、写操作特点,每 Byte 位可以任意由 1 转换为 0,但由 0 转换为 1 时只能按逻辑块为单位,即必须对整个块进行擦除操作,按字节写入数据。

(4) 每块 Flash 的擦除次数是有限制的,NOR Flash 的擦写次数一般为 10~100 万次,但是写入操作则没有次数限制。

因此,文中的 USIM 卡采用的是 NOR Flash。

收稿日期:2009-12-07;修回日期:2010-02-18

基金项目:广东省广州市自然科学基金项目(2008-GX-015)

作者简介:马海峰(1982-),男,河南新野人,硕士研究生,研究方向为智能卡芯片操作系统;李代平,教授,研究方向为智能卡芯片操作系统、网络并行计算。

2 USIM 文件系统的系统结构

2.1 USIM 文件系统的特点

USIM 卡上的文件管理系统和 PC 机上的文件系统相比较,USIM 卡的文件系统具有很多独特的地方。

(1)USIM 卡的文件操作^[4]很简单,不存在拷贝、剪切等。

(2)对于应用来说,其卡内一些文件结构^[5]以及大小都很固定,其存储空间大小和文件的属性在建立文件时就能确定,在使用中不需要修改。

(3)USIM 卡内对文件存储要求空间最优化,系统数据的比重尽可能少。

(4)卡内的文件可以通过文件类型来标识^[4],不同类型的文件对应着不同的操作模式,类型和操作的关联通过文件系统来实现(在 PC 机上一般通过应用程序来实现)。

2.2 USIM 文件系统的系统目标

基于 NOR Flash 的读、写操作特点和保证智能卡文件系统的可靠性,数据的一致性、完整性,以及提高存储空间的利用率对 USIM 卡文件系统要实现的系统目标包括:

(1)碎片空间的回收机制。由于 Flash 存储器在卡的生命周期内要进行文件的创建、更新、删除操作,难免会产生一些碎片,这就需要相应的碎片空间^[6]回收机制,清除废弃的数据来保证存储器的使用效率。碎片空间的回收机制是在创建文件时发生空间不足时被调用。

(2)均衡磨损。针对 Flash 的擦除次数是有限制的,一般支持 10~100 万次的擦除,所以 USIM 文件系统要均衡使用 NOR Flash 的每个扇区,延长 Flash 的使用寿命。

(3)掉电保护机制。由于智能卡 COS 的运行环境非常恶劣,系统资源极其有限,为了保证系统的稳定性、可靠性,要在系统突然断电或重新启动后,能够及时地恢复系统丢失的数据,保证文件系统的一致性、完整性。

2.3 USIM COS 的系统结构模型

USIM COS 系统是具有良好的可维护性、可扩展性的智能卡 COS 系统。系统的总体结构包括业务应用层、功能模块层和微内核层。微内核向上提供统一的硬件驱动接口,直接读、写底层的 Flash 存储数据,功能模块层则实现智能卡 COS 的通信、命令、安全等模块的功能,并向微内核提出服务请求,业务应用层则主要实现基于 USIM COS 开发的应用。COS 微内核对通信管理模块的调度方式采用硬件中断的形式进行调度;功能模块之间的调用采用请求服务/返回响应事

件的方式。USIM COS 的系统模型结构如图 1 所示。

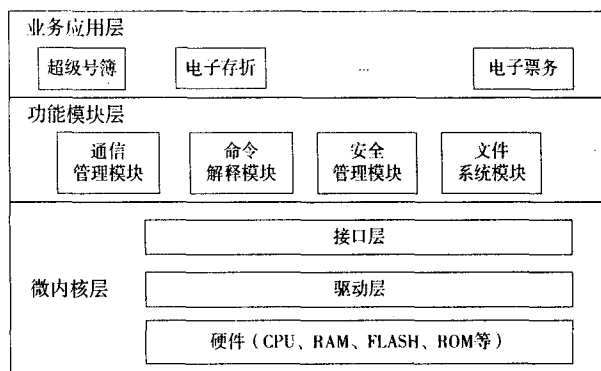


图 1 USIM COS 的系统结构模型图

其中文件管理模块^[2]是智能卡操作系统的核心模块,它负责组织、管理、维护智能卡内存储的所有数据,它是一个承上启下的模块,对底层来说实现存储空间的管理维护,对上层完全透明地实现了数据的管理功能。文件系统要实现的主要功能^[7]:数据有效存储、空间管理、目录文件的查找和定位、文件的创建和删除。文件管理负责数据的组织方式、文件的相关操作;存储管理负责存储空间的分配与回收等操作;命令管理主要实现上层应用。在 ISO/7816-4 中,标准化了智能卡文件系统的逻辑组织结构^[5],要求以文件的形式存储数据,以层次的结构来组织文件的。但对文件的物理存储方式和结构信息没有明确的规定,并且数据的物理存储方式决定了 COS 的性能。因此为了对智能卡内部的文件有效的组织和管理,以及能很好地适应智能卡将来未知文件的产生和新的应用扩充增加。因此文件系统的设计和实现既是 COS 中最灵活的部分,也是直接关系到智能卡的整体性能。

3 USIM 文件系统的存储管理方式

目前 Flash 的存储管理方式主要有:静态存储管理和动态存储管理。

(1)静态存储管理^[8]。在创建文件系统结构时,就根据规范标准各个文件的大小分配数据存储空间,创建 DF、EF 文件^[7]。一旦文件系统通过脚本创建成功后,在整个智能卡操作系统的生命周期内不再更改文件的数据和大小。对文件中的数据进行更新、删除操作都不能大于文件创建时的大小,即整个文件系统的结构和大小是固定不变的。静态存储是依据 NOR Flash 按字节读写的,提高了 Flash 存储空间的读写速度,但是也有很大的缺点,就是容易产生不易回收的碎片。因此静态的存储方式只适合一般的单应用 COS 中,不能适应多应用的需求。

(2)动态存储管理^[8]。动态存储管理是将智能卡

的数据区分为大小相等的块,每块标记一个逻辑块号,再分配空间时根据所需创建文件的大小进行动态的分配。并在创建完文件系统的整体逻辑结构后,还可以根据实际情况再次分配文件的大小。但是逻辑块的大小会影响系统的整体性能,要选择合适的逻辑块大小,一般逻辑块的容量有 64B、128B、256B 三种^[8]。

(3)混合存储管理方式。基于 NOR Flash 以字节为单位写入数据,以块为单位进行擦除的特性,以及为了实现 USIM 卡多应用的动态加载需要动态管理文件的实际情况,在这里采用混合存储管理方式。它是在分配存储空间时,采用静态与动态分配相结合的方式。这种方式兼容了两者的特点,具有较大的灵活性,实现了数据的动态存储。创建目录文件时,只动态分配给该文件一个逻辑块号,将目录文件的大小、类型等信息存储在目录的 FCB 中,其未分配的空间先从当前目录的父目录中把空间预留出来,具体分配那一块,由 Flash 空间中空闲块来决定;而对于 EF 文件,在创建时,一次性连续的分配基本文件所需的存储空间,以后在操作中,不给予分配。此方式存储管理中文件的 FCB 信息由文件名、文件大小、文件描述符、目录项地址、文件其它信息等字段组成,文件的控制信息 FCB 共 20 个字节,其中有几个字节预留处理。

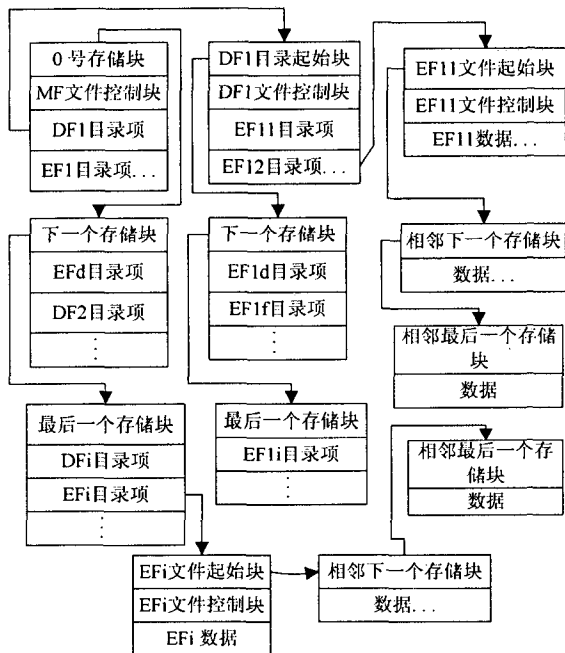


图2 混合存储的数据结构图

智能卡文件系统在创建文件系统结构时,首先创建 MF^[11]文件作为根目录,由系统分配逻辑块,用来存放主文件的 FCB 及 MF 文件下 DF、EF 的目录项^[8];创建 DF 文件时,系统分配的存储空间块用来存放 DF 的 FCB 信息及目录文件下 DF、EF 文件的目录项;创建

EF 文件时,系统分配的存储空间存放 EF 文件的 FCB 信息及 EF 文件数据。其混合存储的数据结构图如图 2 所示。

4 NOR Flash 动态存储管理算法

4.1 存储空间的分配

4.1.1 文件分配表

由于智能卡主要是对于文件系统数据区的分配,文中选择的同方芯片预留 64kB 作为数据存储区,若每个逻辑块大小为 128B,则共 512 块。因此 FAT 表^[9]的大小为 2kB, FAT 表项大小为 4B, FAT 表中用 16bit 的 nextBlock 来存放与该逻辑块相连接的下一个逻辑块号,最后一块的标志是 FFFE,构成数据存储的逻辑地址空间。每个 FAT 表中用 16bit 的 BlockStatus 来表示此逻辑块的使用状态信息,高 8 位记录逻辑块的擦写次数,低 8 位表示逻辑块的状态^[10],其定义如图 3 所示。

状态	定义	描述
FF	空闲块	该块是空闲块,可重新写入数据
FE	有效块	块内容有效且正在使用
FC	垃圾块	块内容处于无效状态

图3 逻辑块状态表

FAT 表的数据结构^[9]如下:

```
typedef struct
{
    iu16 BlockStatus;
    iu16 nextBlock;
} FAT;
```

FAT 表的逻辑结构如图 4 所示。

0x0000	FFFE	FFFE
0x0001	FFFE	0x0002
0x0002	FFFE	0x0003
0x0003	FFFE	0x0005
0x0004	FFFC	0x0007
0x0005	FFFE	FFFE
.....
0x0200	FFFF	FFFF

图4 FAT表的逻辑结构

4.1.2 空间分配的算法

实际上,智能卡的文件系统是比较稳定的,不会发生频繁的建立、删除操作^[1]。对系统来说,真正用于动态分配存储空间的时间只是遍历 FAT 表查找合适大小的空闲块的时间。本算法的空间分配是以块为单位进行分配,只需查找空闲块进行分配,进而提高了算法的效率。其算法如下:

①建立 MF 文件时,首先开始查找 FAT 表是否有空闲块,并判断擦写次数是否是相对较少的。若找到

符合条件的空闲块,移动读指针到 FAT 表空闲表项,并修改 FAT 表信息,否则返回 SW6A84;

②建立 DF、ADF、EF 文件时,判断当前目录是否为其父目录,若不是则选择到其父目录文件^[11];

③读 FAT 表,确定当前目录文件最后所占的块号,计算当前目录下 DF 与 EF 文件的数目;

④是否已经存在同名文件,若是则返回 SW6A89;

⑤若创建的是 DF 文件时,查找空闲目录项,若没有空闲目录项,则执行步骤⑦,否则执行步骤⑧;

⑥若创建的是 EF 文件时,首先读取 FCB 里面文件的大小,若 EF 文件大小 < 逻辑块大小,则实行步骤⑦,否则执行步骤⑧;

⑦读 FAT 表,申请空间,重新分配新块,修改 FAT 表和 FCB 信息;

⑧改变文件系统区域剩余块大小,并填写 FCB 和目录项信息。

4.2 存储空间的回收

随着系统的运行,垃圾块数目不断增加,可用来分配的空闲块越来越少,这就需要系统及时地回收垃圾块,以确保程序的正常运行。为了保证文件系统的一致性和原子性,逻辑块的状态控制信息只能为 FF、FE、FC,即逻辑块的状态要么是空块,要么是有效正在使用,要么就是出于无效状态,不能再有其它状态,易于垃圾块的回收。当一个文件被改写和删除时,所占用的空间并没有立即释放,只是将该文件所占的第一块的状态信息改为 FC 无效状态,并把该文件所分配的第一块逻辑块的逻辑号保存到垃圾块表^[12]中。另外如果突然掉电也会丢失一些数据,造成了数据的不一致性。再次开机时会调用掉电保护恢复数据,有时错误可以通过 CRC 校验恢复,有时是不可恢复的。当恢复操作失败时所产生的垃圾块进行标示,并写入垃圾块表中等待回收。其算法如下:

①当 FAT 表中的空闲块数目小于阈值或创建文件时没有足够的空间分配时调用垃圾块回收函数;

②读取垃圾块表,从中读取一个逻辑块的逻辑号 i;

③计算垃圾块号 i 在 FAT 表中逻辑地址和 Flash 存储器中的物理地址;

④将 FAT 表中的逻辑块 i 状态改为 FF,擦写次数减 1;判断 nextBlock 是否为 FFFE,若是则执行步骤⑤,否则执行步骤⑦;

⑤擦除垃圾块 i;

⑥空闲块数目加 1;

⑦将 $i = \text{nextBlock}$,重复执行步骤 3,4,5,6,直到 nextBlock 的值为 FFFE;

⑧垃圾块数目不为 0 则到步骤②执行,否则继续;

⑨垃圾块回收结束。

4.3 均衡磨损策略

由于 Flash 具有擦出次数限制、先擦后写的特点,会带来使用寿命有限的缺陷^[12]。因此文件系统的设计要考虑均衡地使用 Flash 的每个逻辑块,延长 Flash 的使用寿命,提高 Flash 的使用效率。在设计 FAT 表时就给每个逻辑块分配了 8bit 来记录各个块的擦写次数^[12],初始状态为 FF,共记录次数是 256 次。在创建文件需要重新分配空间时,可以分配擦写次数相对较大的逻辑块。当 Flash 上的某个逻辑块相对其它块的擦写次数低于阈值^[5]时,就说明这个逻辑块上存储的是热数据,这时就要进行均衡处理。将这个逻辑块上的数据迁移到擦写次数较大的空闲块上,然后改写状态为 FC,链接到垃圾块表中。由于分配 FAT 表中的空闲块时是按照擦写次数大的原则分配的,所以低于阈值的逻辑块就很少被分配出去,只有当 FAT 表中存在大量的逻辑块的擦写次数低于阈值时,就要重新将 FAT 表中的所有逻辑块的状态擦写为 FF。重复如此操作,从一定程度上均衡使用了各个逻辑块,起到了尽量延长其使用寿命的目的。

5 结束语

文中所介绍的 USIM 文件系统中存储空间的动态管理方法能很好地利用智能卡上有限的存储空间,且可以将文件系统中的数据存放于一块连续的存储空间中,减少了在查找和更新数据上所花的时间。介绍了一种碎片空间的回收机制和均衡磨损^[12]的算法,最大化地延长 Flash 存储器的使用寿命,提高存储器的使用效率。这对智能卡的操作系统的设计有一定的指导意义。

参考文献:

- [1] 刘刚,李铮,杨先文.智能卡应用框架的研究及应用[J].计算机工程与设计,2007,28(23):5784-5786.
- [2] 张勇,裘雪红.嵌入式 Linux 下 JFFS2 文件系统的实现[J].计算机技术与发展,2006,16(4):138-140.
- [3] 马丰玺,杨斌.非易失存储器 NAND Flash 及其在嵌入式系统中的应用[J].计算机技术与发展,2007,17(1):203-204.
- [4] ETSI TS 102.222 Smart Cards, UICC-Terminal interface; Physical and logical characteristics[M]. [s.l.]:[s.n.],2007.
- [5] Seung-Ho Lim, Kyu-Ho Park. An Efficient NAND Flash File System for Flash Memory Storage[J]. IEEE Transactions on Computers,2006,55(7):906-912.

(下转第 78 页)

本算法自身特点决定的。

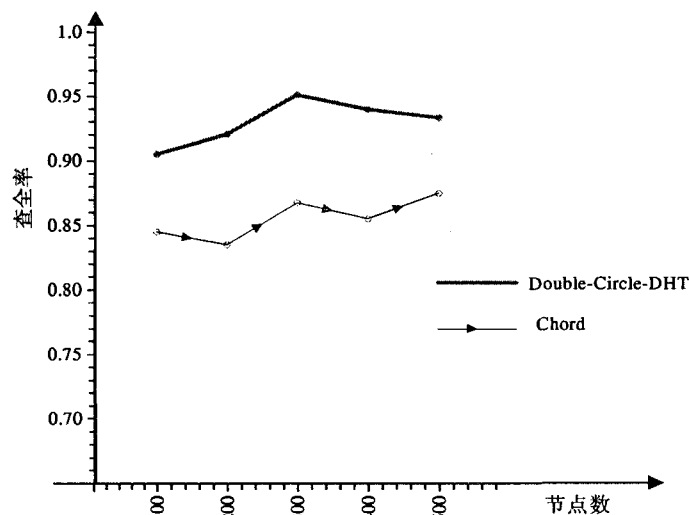


图 3 Double-Circle-DTH 算法
和 Chord 算法查全率

由图 3 可知, Chord 算法的查全率在 80%~90% 之间, 而本算法查全率则始终保持在 90% 以上, 主要是因为文中采用的算法增加了基于语义的模糊匹配, 在传统 Chord 算法查询失败后进行语义匹配并返回语义最相似的结果, 因此提高了查全率。

以上实验证明, 文中采用的 Double-Circle-DTH 搜索算法通过实现精确查找和模糊匹配的融合, 从而实现查全率的绝对优势, 但是由于算法自身的特点决定了本算法的搜索平均跳数和平均延时高于 Chord 算法。随着网络规模的增大, Double-Circle-DTH 算法的搜索平均跳数则有明显高于 Chord 算法的趋势。

4 结束语

在基于精确查找的 P2P 搜索算法基础上, 通过引入基于语义划分的模糊匹配提高搜索的命中率。通过仿真实验表明 Double-Circle-DTH 算法在小规模稳定的网络环境中能够以相对稍多的开销实现绝对优势的高命中率, 当网络规模增大时则显露出其高开销的劣势。

下一步的工作是对 Double-Circle-DTH 算法进行优化, 降低搜索的开销, 提高搜索的效率, 使其能够适应大规模的复杂网络环境。

参考文献:

- [1] Stoica I, Morrisr, Uben - Nowell D, et al. Chord: a scalable peer-to-peer lookup protocol for Internet applications[J]. IEEE//ACM Trans on Networking, 2003, 11(1): 17-32.
- [2] Zhao B Y, Ling Huang, Strib Ling J, et al. Tapestry: a resilient global-scale overlay for service deployment [J]. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 41-53.
- [3] Ratnasamy S, Francis P, Handley M, et al. A Scalable Content-addressable Network [C]//Proc. of ACM SIGCOMM. [s.l.]: [s.n.], 2001.
- [4] Rowstron A, Drusche P. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems[EB/OL]. 2001. <http://research.microsoft.com/antr/pastry/>, 2001: 188-192.
- [5] Berners-Lee T, Hendler J, Lassila O. The semantic web[EB/OL]. 2001-05. <http://www.sciam.com>.
- [6] Salton G, Wong A. A vector space model for automatic indexing[J]. Communications of ACM, 1975, 18(11): 613-620.
- [7] 蔡诗维, 钟 诚. 基于兴趣社群的 P2P 网络节点自治管理方法[J]. 计算机技术与发展, 2008, 18(3): 161-164.
- [8] Salton G, Yang C. On the specification of term values in automatic indexing[J]. Journal of Documentation, 1973, 2(94): 351-372.
- [9] 赵 宏, 谢伟志, 张晨曦. 基于蚁群算法的非结构化 P2P 搜索研究[J]. 计算机技术与发展, 2009, 19(2): 31-34.
- [10] 李 想, 吴国新, 郭 晶. 基于分布式倒排索引和 VSM 算法的 P2P 复杂搜索[J]. 计算机技术与发展, 2009, 19(4): 25-27.
- [11] Salton G, Lesk M. Computer evaluation of indexing and text processing[J]. Journal of ACM, 1968, 15(1): 8-36.
- [12] 王志晓, 张大陆, 刘 雷, 等. 支持语义的 P2P 搜索研究[J]. 计算机工程与应用, 2007, 43(3): 8-11.

(上接第 74 页)

- [6] 沈建华, 罗悦泽. Flash 文件系统的研究与设计[J]. 计算机应用研究, 2004, 12(3): 246-248.
- [7] 李 翔. 智能卡研发技术与工程实践[M]. 北京: 中国书籍出版社, 2003: 61-202.
- [8] 张鲁国, 马自堂. 智能卡操作系统中存储管理设计[J]. 微计算机信息, 2005, 21(9-3): 22-23.
- [9] 吴 瀛, 龚育昌, 张为民. 一种基于 Flash 的智能卡文件系统[J]. 计算机工程, 2005, 31(6): 55-56.
- [10] 王新华, 王建芬. NAND Flash 在嵌入式文件系统中的一种动态管理方法[J]. 计算机应用与软件, 2008, 25(9): 95-98.
- [11] ETSI TS 102. 221 UICC - Terminal interface, Physical and logical characteristics[M]. [s.l.]: [s.n.], 2007.
- [12] 潘 沁, 周新志, 魏 刚. 磨损均衡算法在 NAND Flash 管理中的改进[J]. 微计算机信息, 2007, 23(3-1): 301-302.