```
     error=15078 txt: ''
Automatic datafile offline due to write error on
file 57: +DG_DATA_03/billbj/datafile/tbs_band_dailytable_20.273.656599591
KCF: write/open error block=0x8e20b online=1
     file=68 +DG_DATA_03/billbj/datafile/tbs_band_table_idx_20.281.659614825
     error=15078 txt: ''
```

此时检查 ASM 的告警日志，也可以发现 ASM 实例无法加载磁盘组（DG_DATA_03）的错误：

```
Thu Jun 25 05:10:49 2009
NOTE: recovering COD for group 1/0x6808040 (DG_DATA_01)
SUCCESS: completed COD recovery for group 1/0x6808040 (DG_DATA_01)
NOTE: recovering COD for group 2/0x6908041 (DG_DATA_02)
SUCCESS: completed COD recovery for group 2/0x6908041 (DG_DATA_02)
Thu Jun 25 05:12:59 2009
Errors in file /u01/app/oracle/admin/+ASM/bdump/+asm1_gmon_21761.trc:
ORA-27091: unable to queue I/O
ORA-27072: File I/O error
HPUX-ia64 Error: 6: No such device or address
Additional information: 4
Additional information: 2044
Additional information: -1
Thu Jun 25 05:13:29 2009
WARNING: cache failed to read fn=3 blk=0 from disk(s): 1
ORA-15062: ASM disk is globally closed
NOTE: cache initiating offline of disk 1  group 3
Thu Jun 25 05:22:57 2009
ERROR: no PST quorum in group 3: required 1, found 0
Thu Jun 25 05:22:57 2009
Errors in file /u01/app/oracle/admin/+ASM/bdump/+asm1_n000_28730.trc:
ORA-15063: ASM discovered an insufficient number of disks for diskgroup "DG_DATA_03"
Thu Jun 25 05:22:57 2009
ERROR: async update- could not update PST (grp 3)
Thu Jun 25 05:22:57 2009
Errors in file /u01/app/oracle/admin/+ASM/bdump/+asm1_n000_28730.trc:
```

```
ORA-15063: ASM discovered an insufficient number of disks for diskgroup "DG_DATA_03"
```

当用户解决了交换机问题之后，试图恢复数据库运行时，遇到了另外一个问题，从数据库角度看来，始终有一个磁盘组处于 Mounted 的状态，数据库无法连接，也就无法访问其中的数据：

```
SQL> select name,state from v$asm_diskgroup_stat;
NAME                            STATE
------------------------------ -----------
DG_DATA_01                      CONNECTED
DG_DATA_02                      CONNECTED
DG_DATA_03                      MOUNTED
```

并且从告警日志来看，ASM 磁盘组在挂载后会很快被自动卸载：

```
Thu Jun 25 17:45:11 2009
SUCCESS: diskgroup DG_DATA_03 was mounted
SUCCESS: diskgroup DG_DATA_03 was dismounted
SUCCESS: diskgroup DG_DATA_03 was mounted
SUCCESS: diskgroup DG_DATA_03 was dismounted
SUCCESS: diskgroup DG_DATA_03 was mounted
SUCCESS: diskgroup DG_DATA_03 was dismounted
```

如何修正磁盘组的状态成为了一个难题，如果该磁盘组无法连接，则数据库就无法访问其中的数据。

恢复过程

我们尝试直接拷贝复制磁盘组中的文件，发现可以成功，而在文件备份过程中，磁盘组的状态在数据库中转为正常的 CONNECTED 模式：

```
oracle@ccnbjdc1[billbj1]:/u01/app/oracle/admin/billbj/bdump$ rman target /
Recovery Manager: Release 10.2.0.3.0 - Production on Thu Jun 25 17:45:00 2009
Copyright (c) 1982, 2005, Oracle.  All rights reserved.
connected to target database: BILLBJ (DBID=2424179062)
RMAN> copy datafile '+DG_DATA_03/billbj/datafile/tbs_default_20.264.654269073' to '/backup/a.dbf';
Starting backup at 25-JUN-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: sid=1152 instance=billbj1 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=1155 instance=billbj1 devtype=DISK
allocated channel: ORA_DISK_3
channel ORA_DISK_3: sid=1153 instance=billbj1 devtype=DISK
allocated channel: ORA_DISK_4
channel ORA_DISK_4: sid=1150 instance=billbj1 devtype=DISK
channel ORA_DISK_1: starting datafile copy
input datafile fno=00161
name=+DG_DATA_03/billbj/datafile/tbs_default_20.264.654269073
output filename=/backup/a.dbf tag=TAG20090625T174511 recid=2 stamp=690486558
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:04:09
Finished backup at 25-JUN-09

Starting Control File and SPFILE Autobackup at 25-JUN-09
piece handle=/backup/ctlbak/c-2424179062-20090625-01 comment=NONE
Finished Control File and SPFILE Autobackup at 25-JUN-09
```

至此，结合前面分析的文件 Offline 状态，我们得出以下结论：

当磁盘组中的所有文件 Offline，则 Oracle 不访问该 DG 中的磁盘，该磁盘就保持了 MOUNTED 状态，数据库无需连接磁盘组；我们只要尝试访问该磁盘中的文件，该磁盘组就会显示为数据库连接的 CONNECTED 状态。

接下来通过 Recover 那些被 Offline 的文件，再执行 Online 操作，就将数据库恢复到了正常状态。

```
Thu Jun 25 17:54:22 2009
ALTER DATABASE RECOVER  datafile 151
Thu Jun 25 17:54:22 2009
Media Recovery Start
 parallel recovery started with 3 processes
ORA-279 signalled during: ALTER DATABASE RECOVER  datafile 151  ...
Thu Jun 25 17:54:24 2009
ALTER DATABASE RECOVER    CONTINUE DEFAULT
Thu Jun 25 17:54:24 2009
Media Recovery Log +DG_DATA_02/billbj/1_134766_634298105.dbf
```

```
ORA-279 signalled during: ALTER DATABASE RECOVER    CONTINUE DEFAULT  ...
Thu Jun 25 17:54:24 2009
ALTER DATABASE RECOVER    CONTINUE DEFAULT
Thu Jun 25 17:54:24 2009
Media Recovery Log +DG_DATA_02/billbj/2_90114_634298105.dbf
ORA-279 signalled during: ALTER DATABASE RECOVER    CONTINUE DEFAULT  ...
Thu Jun 25 17:54:25 2009
ALTER DATABASE RECOVER    CONTINUE DEFAULT
Thu Jun 25 17:54:25 2009
Media Recovery Log +DG_DATA_02/billbj/2_90115_634298105.dbf
ORA-279 signalled during: ALTER DATABASE RECOVER    CONTINUE DEFAULT  ...
Thu Jun 25 17:54:27 2009
ALTER DATABASE RECOVER    CONTINUE DEFAULT
Thu Jun 25 17:54:27 2009
Media Recovery Log +DG_DATA_02/billbj/1_134767_634298105.dbf
Thu Jun 25 17:54:28 2009
Media Recovery Complete (billbj1)
```

这个数据库是一个 TB 级别的核心计费数据库，非常重要：

```
SQL> select sum(bytes)/1024/1024/1024 from v$datafile;
SUM(BYTES)/1024/1024/1024
-------------------------
               1046.01709
```

因为简单的一次维护碰撞，业务遭受了一天的影响。

又一次碰撞引发的灾难

文件离线与归档缺失案例

前面提到的用户属于幸运者，如果用户不能及时发现和解决这个问题，如果不是一个 ASM 磁盘组的整体问题，也许故障就会被掩盖，灾难就会更惨重。

从一个侧面来说，Oracle 的自动数据库文件离线保护机制是有问题的，或者说提示不够明确，数据库应当在启动过程中，予以明确提示用户，部分文件因为保护离线，请用户处理，如果能够在这个环境做出提示，本章的几个案例就都会大大简化。

灾难描述

以下是与上一节完全类似的一个案例，但是错误走得更远，灾难也就更加严重:

1. 集成商为用户扩展存储，增加硬盘
2. 无意中将光线交换机碰断电
3. 部分数据库文件出现读写错误离线
4. 重启数据库后未察觉
5. 多日后发现，执行在线恢复
6. 发现丢失了归档日志，数据文件无法加载
7. 灾难形成

丢失了归档日志，文件离线，使得这个案例变得异常复杂。

案例警示

这个案例给我们的警示有：

1. **墨菲定律总是无处不在**

墨菲定律告诉我们，你越是害怕出现问题的地方，就越是会出现问题；哪怕你觉得需要 N 多个条件才能出

现的问题，这 N 多个条件终将满足。

所以不能在任何一个环节掉以轻心，我们举一个管理学中常见的比喻：假定一个故障在可靠性低于 60% 才能发生，我们认为这个概率已经很低；另一个条件，通常我们觉得将一个工作的准确性达到 90%已经足够优秀；现在的问题时，如果多个环节都做到 90%会怎样？

假定数据环境涉及到了网络、主机、存储、操作系统和数据库五大环节，如果每个环境都做到了 90 分，那么最后整个系统的可靠性有多少呢？

90% × 90% × 90% × 90% × 90%=59.049%

最终答案是 59.049%,这个指标已经低于了 60%，这个系统的稳定性将会出现问题。所以在我们经手的每一个环境，都应当力争做到 100 分，这样才能为其他环境留下机会，为系统的稳健提供保护。

2. 在维护工作之后进行日志检查

通常维护工作都在深夜来完成，而人在疲劳加班之后，潜意识会想要尽快完成工作，离开现场，这就为工作留下了隐患。

根据我们的经验，在维护时出现的问题，通过日志监控和检查都可以发现。

我们建议，**对于数据库环境，在维护期间应当提炼摘取维护期生成的所有日志，确保日志中没有出现错误，或者出现的错误都得到了处理**。

这样至少可以避免多数基本故障。

3. 不要过分信赖数据库的自我修复能力

对于类似这样的故障，很多用户认为，Oracle 应当能够通过自我调整来完成故障恢复，但是显然，我们不能对数据库要求太高。在这个案例中，保护性离线之后，Oracle 不会自动进行文件恢复和在线尝试（实际上这种情况是应当可以自动修复的），如果用户疏忽，则故障就可能出现。

对于自动保护的文件离线问题，实际上我认为是 Oracle 数据库的 BUG，Oracle **应当能够分辨哪些情况文件是由于保护方式离线的，并且应当在数据库启动之后，给予用户强制性提示，要求用户进行判断和处理**，如果这样，本章所描述的案例就不会触发如此复杂严重的事故。

技术和管理相结合，才能确保数据库的安全。

技术回放

在检查用户数据库告警日志时，首先发现客户数据库早就存在 ORA-600 错误：

```
Fri Mar 11 01:58:11 2011
Errors in file /opt/oracle/admin/orcl/udump/orcl_ora_233946.trc:
ORA-00600: internal error code, arguments: [25012], [1], [2], [], [], [], [], []
ARC0: Completed archiving  log 5 thread 1 sequence 81253
ARC0: Evaluating archive   log 7 thread 2 sequence 27079
ARC0: Beginning to archive log 7 thread 2 sequence 27079
Creating archive destination LOG_ARCHIVE_DEST_1: '/arch/2_27079.dbf'
Fri Mar 11 01:58:11 2011
Errors in file /opt/oracle/admin/orcl/udump/orcl_ora_233946.trc:
ORA-00600: internal error code, arguments: [25012], [1], [2], [], [], [], [], []
```

数据库的参数文件中，设置了大量隐含参数进行错误屏蔽：

```
  db_file_multiblock_read_count= 16
  fast_start_mttr_target   = 300
  rollback_segments        = system
  _corrupted_rollback_segments= _SYSSMU1$, _SYSSMU2$, _SYSSMU3$, _SYSSMU4$, _SYSSMU5$,
_SYSSMU6$, _SYSSMU7$, _SYSSMU8$, _SYSSMU9$, _SYSSMU10$, _SYSSMU11$, _SYSSMU12$,
_SYSSMU13$, _SYSSMU14$, _SYSSMU15$, _SYSSMU16$, _SYSSMU17$, _SYSSMU18$, _SYSSMU19$,
_SYSSMU20$
  undo_management          = MANUAL
  undo_tablespace          = UNDOTBS4
  undo_retention           = 10800
  remote_login_passwordfile= EXCLUSIVE
```

这其中的部分参数设置存在问题，另外，一个生产数据库，是不应该在这些参数的保护下运行的，如果错误解决不能排除所有异常，那么数据库就应当重建。

在客户交换机问题出现时，数据库出现如下错误，文件无法读写：

```
Sun Jul 24 10:09:24 2011
Errors in file /opt/oracle/admin/orcl/udump/orcl_ora_618684.trc:
ORA-01115: IO error reading block from file 72 (block # 500191)
```

```
ORA-27063: skgfospo: number of bytes read/written is incorrect
IBM AIX RISC System/6000 Error: 5: I/O error
Additional information: -1
Additional information: 32768
ORA-01115: IO error reading block from file 72 (block # 500191)
ORA-27063: skgfospo: number of bytes read/written is incorrect
IBM AIX RISC System/6000 Error: 5: I/O error
Additional information: -1
Additional information: 32768
ORA-01115: IO error reading block from file 72 (block # 500191)
ORA-27063: skgfospo: number of bytes read/written is incorrect
IBM AIX RISC System/6000 Error: 5: I/O error
Additional information: -1
Additional information: 32768
```

接下来数据库保护性的将发生读写错误的数据文件离线：

```
Sun Jul 24 10:31:56 2011
KCF: write/open error block=0x1f1d71 online=1
     file=36 /dev/ro_nlv_img_08
     error=27063 txt: 'IBM AIX RISC System/6000 Error: 5: I/O error
Additional information: -1
Additional information: 8192'
Sun Jul 24 10:31:56 2011
KCF: write/open error block=0xa2b online=1
     file=82 /dev/ro_dt_vio_index_
     error=27063 txt: 'IBM AIX RISC System/6000 Error: 5: I/O error
Additional information: -1
Additional information: 8192'
Sun Jul 24 10:31:56 2011
Errors in file /opt/oracle/admin/orcl/udump/orcl_ora_278924.trc:
ORA-01115: IO error reading block from file 58 (block # 1020235)
ORA-27063: skgfospo: number of bytes read/written is incorrect
IBM AIX RISC System/6000 Error: 5: I/O error
Additional information: -1
```

```
Additional information: 8192
Automatic datafile offline due to write error on
file 36: /dev/ro_nlv_img_08
Sun Jul 24 10:31:57 2011
Automatic datafile offline due to write error on
file 82: /dev/ro_dt_vio_index_
```

在修复了交换机问题之后，用户将数据库启动：

```
Sun Jul 24 11:39:22 2011
ALTER DATABASE OPEN
Sun Jul 24 11:39:23 2011
Beginning crash recovery of 1 threads
Sun Jul 24 11:39:23 2011
Started first pass scan
Sun Jul 24 11:39:24 2011
Completed first pass scan
 160936 redo blocks read, 2669 data blocks need recovery
>>数据库执行恢复
Sun Jul 24 11:39:24 2011
Started recovery at
 Thread 1: logseq 120428, block 177143, scn 0.0
Recovery of Online Redo Log: Thread 1 Group 1 Seq 120428 Reading mem 0
  Mem# 0 errs 0: /dev/ro_log1_01
Recovery of Online Redo Log: Thread 1 Group 5 Seq 120429 Reading mem 0
  Mem# 0 errs 0: /dev/ro_log1_03
Sun Jul 24 11:39:28 2011
Completed redo application
>>数据库完成恢复
Sun Jul 24 11:39:30 2011
Ended recovery at
 Thread 1: logseq 120429, block 133288, scn 2868.1571390680
 2669 data blocks read, 2669 data blocks written, 160936 redo blocks read
Crash recovery completed successfully
Sun Jul 24 11:39:31 2011
```

```
LGWR: Primary database is in CLUSTER CONSISTENT mode
Thread 1 advanced to log sequence 120430
Thread 1 opened at log sequence 120430
  Current log# 2 seq# 120430 mem# 0: /dev/ro_log1_02
Successful open of redo thread 1.
Sun Jul 24 11:39:31 2011
SMON: enabling cache recovery
>>数据库开始工作归档
Sun Jul 24 11:39:31 2011
ARC0: Evaluating archive   log 5 thread 1 sequence 120429
ARC0: Beginning to archive log 5 thread 1 sequence 120429
Creating archive destination LOG_ARCHIVE_DEST_1: '/arch/1_120429.dbf'
ARC0: Completed archiving  log 5 thread 1 sequence 120429
```

注意，此时用户认为数据库已经恢复了正常，能够提供服务，开始切换归档日志。但是没有人注意到，有几个数据库文件已经 Offline 离线。

等到用户注意到这个问题，尝试去加载这些文件时，发现归档日志丢失了，没有了归档日志，这些数据文件无法被 Online：

```
Mon Jul 25 14:25:12 2011
ALTER DATABASE RECOVER  datafile '/dev/ro_dt_vio_dat_02'
Mon Jul 25 14:25:12 2011
Media Recovery Datafile: '/dev/ro_dt_vio_dat_02'
Media Recovery Start
Starting datafile 94 recovery in thread 1 sequence 120428
Datafile 94: '/dev/ro_dt_vio_dat_02'
Media Recovery Log
ORA-279 signalled during: ALTER DATABASE RECOVER  datafile '/dev/ro_dt_vio_d...
Mon Jul 25 14:25:15 2011
ALTER DATABASE RECOVER    CONTINUE DEFAULT
Media Recovery Log /arch/1_120428.dbf
Errors with log /arch/1_120428.dbf.
ORA-308 signalled during: ALTER DATABASE RECOVER    CONTINUE DEFAULT  ...
Mon Jul 25 14:25:15 2011
```

```
ALTER DATABASE RECOVER    CONTINUE DEFAULT
Media Recovery Log /arch/1_120428.dbf
Errors with log /arch/1_120428.dbf.
ORA-308 signalled during: ALTER DATABASE RECOVER    CONTINUE DEFAULT  ...
Mon Jul 25 14:25:15 2011
ALTER DATABASE RECOVER CANCEL
Media Recovery Cancelled
Completed: ALTER DATABASE RECOVER CANCEL
```

这里的错误 ORA-279 在前台出现的错误提示就是归档日志不可用：

```
[oracle@hpserver2 ~]$ oerr ora 279
00279, 00000, "change %s generated at %s needed for thread %s"
// *Cause: The requested log is required to proceed with recovery.
// *Action:  Please supply the requested log with "ALTER DATABASE RECOVER
//           LOGFILE <file_name>" or cancel recovery with "ALTER DATABASE
//           RECOVER CANCEL".
```

用户数据库大小近 3TB，包含了长期以来的数据积累：

```
select sum(bytes)/1024/1024/1024 GB from v$datafile;
        GB
----------
2816.17435
```

离线的文件主要有以下系列：

```
select file#,name,bytes/1024/1024 MB,status from v$datafile where status like 'REC%';
     FILE# NAME                                MB STATUS
---------- ------------------------- ---------- -------
        36 /dev/ro_nlv_img_08        32767.9922 RECOVER
        82 /dev/ro_dt_vio_index_           4000 RECOVER
        94 /dev/ro_dt_vio_dat_02          10000 RECOVER
        95 /dev/ro_dt_vio_dat_03          10000 RECOVER
        96 /dev/ro_dt_vio_dat_04          10000 RECOVER
```

对于丢失了归档日志文件的情况，正常情况下，Oracle 不允许跳过归档加载文件，因为丢失归档日志意味着数据库的一致性被破坏，应当通过备份来恢复数据。

但是如果没有备份，我们就只能通过特殊的手段来进行恢复尝试，这种尝试仅在迫不得已的情况下使用，并且应当在之后重建数据库。

恢复过程

以下是本书提供的一个恢复测试说明。

1.BBED 修改文件头跳过归档日志

首先创建一个包含两个数据文件的表空间 USERS：

```
SQL> alter database datafile 'C:\ORACLE\ORADATA\ORA9I\USERS01.DBF' resize 2M;
Database altered.
SQL> select name,bytes/1024/1024 from v$datafile;
NAME                                               BYTES/1024/1024
-------------------------------------------------- ---------------
C:\ORACLE\ORADATA\ORA9I\SYSTEM01.DBF                           250
C:\ORACLE\ORADATA\ORA9I\UNDOTBS01.DBF                          200
C:\ORACLE\ORADATA\ORA9I\USERS01.DBF                              2
SQL> alter tablespace users add datafile 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF' size 2M;
Tablespace altered.
SQL> alter database datafile 'C:\ORACLE\ORADATA\ORA9I\USERS01.DBF' autoextend off;
Database altered.
```

在表空间创建一个数据表，使用完所有的空间：

```
SQL> create table eygle tablespace users as select * from dba_objects;
Table created.
SQL> insert into eygle select * from eygle;
6323 rows created.
SQL> insert into eygle select * from eygle;
insert into eygle select * from eygle
            *
ERROR at line 1:
```

```
ORA-01653: unable to extend table SYS.EYGLE by 128 in tablespace USERS
SQL> select count(*) from eygle;
  COUNT(*)
----------
     12646
```

确保数据库运行在归档模式下，将数据文件离线：

```
SQL> archive log list;
Database log mode              Archive Mode
Automatic archival             Disabled
Archive destination            c:\oracle\9.2.0\RDBMS
Oldest online log sequence     72
Next log sequence to archive   74
Current log sequence           74
SQL> archive log start;
Statement processed.
SQL> alter database datafile 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF' offline;
Database altered.
```

切换一些归档，然后删除这些归档日志：

```
SQL> alter system switch logfile;
System altered.
SQL> alter system switch logfile;
System altered.
SQL> alter system switch logfile;
System altered.
SQL> select file#,status from v$datafile;
     FILE# STATUS
---------- -------
         1 SYSTEM
         2 ONLINE
         3 ONLINE
         4 RECOVER
```

如果此时尝试 Online 数据文件，会要求进行恢复，如果无法找到需要的归档日志，则恢复无法进行，这就是用户面对的情况：

```
SQL> alter database datafile 4 online;
alter database datafile 4 online
*
ERROR at line 1:
ORA-01113: file 4 needs media recovery
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
SQL> recover datafile 4;
ORA-00279: change 223897 generated at 01/11/2012 16:19:23 needed for thread 1
ORA-00289: suggestion : C:\ORACLE\9.2.0\RDBMS\ARC00074.001
ORA-00280: change 223897 for thread 1 is in sequence #74


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00308: cannot open archived log 'C:\ORACLE\9.2.0\RDBMS\ARC00074.001'
ORA-27041: unable to open file
OSD-04002: 无法打开文件
O/S-Error: (OS 2) The system cannot find the file specified.
SQL> select file#,name from v$datafile;
     FILE# NAME
---------- ----------------------------------------
         1 C:\ORACLE\ORADATA\ORA9I\SYSTEM01.DBF
         2 C:\ORACLE\ORADATA\ORA9I\UNDOTBS01.DBF
         3 C:\ORACLE\ORADATA\ORA9I\USERS01.DBF
         4 C:\ORACLE\ORADATA\ORA9I\USERS02.DBF
```

解决这个问题的一个办法是通过 BBED 来进行数据文件头，跳过缺失的归档日志文件，然后强制挂接数据文件。以下是详细的过程介绍。

BBED 的基本配置

首先编辑好 BBED 需要的参数文件，一个用于提供文件列表，一个用于设定基本参数：

```
E:\>type data.txt
         1 C:\ORACLE\ORADATA\ORA9I\SYSTEM01.DBF
         2 C:\ORACLE\ORADATA\ORA9I\UNDOTBS01.DBF
         3 C:\ORACLE\ORADATA\ORA9I\USERS01.DBF
         4 C:\ORACLE\ORADATA\ORA9I\USERS02.DBF
E:\>type par.txt
listfile=data.txt
mode=edit
blocksize=8192
```

使用如下命令启动 BBED 程序：

```
E:\>bbed parfile=par.txt
```

最简单的，由于文件 3 和 4 属于同一个表空间，3 号文件一切完好，我们可以将 3 号文件的头块覆盖到 4 号文件的文件头上。

BBED COPY 进行块复制恢复

BBED 的 COPY 命令语法如下：

```
COPY [ DBA | FILE | FILENAME | BLOCK ] TO [ DBA | FILE | FILENAME | BLOCK ]
```

以下命令将文件 3 的头块拷贝复制到文件 4 的头块上：

```
BBED> copy file 3 block 1 to file 4 block 1;
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:    0 to    7           Dba:0x01000001
------------------------------------------------------------------------
 0b020000 0100c000

 <32 bytes per line>
BBED> set file 4 block 1;
```

```
        FILE#           4
        BLOCK#          1
BBED> dump
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:    0 to    7           Dba:0x01000001
------------------------------------------------------------------------
 0b020000 0100c000


 <32 bytes per line>
```

数据文件头的信息可以通过 map 命令展示出来，以下输出显示数据文件头主要的数据结构为 kcvfh，共占用 360 字节存储，右侧显示的是具体内容的偏移量：

```
BBED> map /v
 File:  (0)
 Block: 1                                      Dba:0x00000000
------------------------------------------------------------
 Data File Header

 struct kcvfh, 360 bytes                    @0
    struct kcvfhbfh, 20 bytes               @0
    struct kcvfhhdr, 76 bytes               @20
    ub4 kcvfhrdb                            @96
    struct kcvfhcrs, 8 bytes                @100
    ub4 kcvfhcrt                            @108
    ub4 kcvfhrlc                            @112
    struct kcvfhrls, 8 bytes                @116
    ub4 kcvfhbti                            @124
    struct kcvfhbsc, 8 bytes                @128
    ub2 kcvfhbth                            @136
    ub2 kcvfhsta                            @138
    struct kcvfhckp, 36 bytes               @140
    ub4 kcvfhcpc                            @176
    ub4 kcvfhrts                            @180
    ub4 kcvfhccc                            @184
```

```
    struct kcvfhbcp, 36 bytes              @188
    ub4 kcvfhbhz                           @224
    struct kcvfhxcd, 16 bytes              @228
    word kcvfhtsn                          @244
    ub2 kcvfhtln                           @248
    text kcvfhtnm[30]                      @250
    ub4 kcvfhrfn                           @280
    struct kcvfhrfs, 8 bytes               @284
    ub4 kcvfhrft                           @292
    struct kcvfhafs, 8 bytes               @296
    ub4 kcvfhbbc                           @304
    ub4 kcvfhncb                           @308
    ub4 kcvfhmcb                           @312
    ub4 kcvfhlcb                           @316
    ub4 kcvfhbcs                           @320
    ub2 kcvfhofb                           @324
    ub2 kcvfhnfb                           @326
    ub4 kcvfhprc                           @328
    struct kcvfhprs, 8 bytes               @332
    struct kcvfhprfs, 8 bytes              @340
    ub4 kcvfhtrt                           @356

 ub4 tailchk                               @8188
```

使用 print 命令可以具体打印出相关变量的信息：

```
BBED> p kcvfh
struct kcvfh, 360 bytes                 @0
   struct kcvfhbfh, 20 bytes            @0
      ub1 type_kcbh                        @0        0x0b
      ub1 frmt_kcbh                        @1        0x02
      ub1 spare1_kcbh                      @2        0x00
      ub1 spare2_kcbh                      @3        0x00
      ub4 rdba_kcbh                        @4        0x01000001
      ub4 bas_kcbh                         @8        0x00000000
```

```
      ub2 wrp_kcbh                            @12       0x0000
      ub1 seq_kcbh                            @14       0x01
      ub1 flg_kcbh                            @15       0x04 (KCBHFCKV)
      ub2 chkval_kcbh                         @16       0x31d1
      ub2 spare3_kcbh                         @18       0x0000
```

拷贝之后还有几个内容需要修改，主要是文件号相关的信息。偏移量 4 记录的是 RDBA 信息，其中包含文件号信息，此处需要根据情况由 3 改为 4：

```
BBED> set offset 4
        OFFSET          4
BBED> dump
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:    4 to   11           Dba:0x01000001
------------------------------------------------------------------------
 0100c000 00000000

 <32 bytes per line>
BBED> modify /x 01000001
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:    4 to   11           Dba:0x01000001
------------------------------------------------------------------------
 01000001 00000000

 <32 bytes per line>
```

此外偏移量 52 处存储的是文件号信息：

```
      ub2 kccfhfno                            @52       0x0004
      ub2 kccfhtyp                            @54       0x0003
```

也需要同样修改：

```
BBED> set offset 52
        OFFSET          52
BBED> dump
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
```

```
 Block: 1                Offsets:   52 to   59           Dba:0x01000001
------------------------------------------------------------------------
 03000300 00000000

 <32 bytes per line>
BBED> modify /x 04
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:   52 to   59           Dba:0x01000001
------------------------------------------------------------------------
 04000300 00000000

 <32 bytes per line>
```

偏移量 280 处存储的是相对文件号：

```
   ub4 kcvfhrfn                            @280      0x00000004
   struct kcvfhrfs, 8 bytes                @284
      ub4 kscnbas                          @284      0x00000000
      ub2 kscnwrp                          @288      0x0000
   ub4 kcvfhrft                            @292      0x2e086f78
```

以下需要相应的修改：

```
BBED> set offset 280
        OFFSET          280
BBED> dump
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:  280 to  287           Dba:0x01000001
------------------------------------------------------------------------
 03000000 00000000

 <32 bytes per line>
BBED> modify /x 04
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:  280 to  287           Dba:0x01000001
------------------------------------------------------------------------
```

```
 04000000 00000000

 <32 bytes per line>
```

BBED> sum apply

数据文件创建时间与 SCN 校验

如果此时尝试恢复数据文件会遇到如下错误，提示文件 4 的创建 SCN 错误：

SQL> recover datafile 4;

```
ORA-00283: recovery session canceled due to errors
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
ORA-01122: database file 4 failed verification check
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
ORA-01203: wrong incarnation of this file - wrong creation SCN
```

数据文件的 SCN 来自数据字典 file$视图：

SQL> select file#,crscnwrp,crscnbas,to_char(crscnbas,'xxxxxx') scn from file$;

```
     FILE#   CRSCNWRP   CRSCNBAS SCN
---------- ---------- ---------- -------
         1          0          9       9
         2          0       4480    1180
         3          0       5812    16b4
         4          0     222765   3662d
```

数据文件的创建 SCN 存储与偏移量 100 处：

```
   struct kcvfhcrs, 8 bytes                @100
      ub4 kscnbas                          @100      0x0003662d
      ub2 kscnwrp                          @104      0x0000
```

根据这个文件的具体信息，修改这个 SCN：

BBED> set offset 100

```
        OFFSET          100
```

BBED> dump

```
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
```

```
 Block: 1                Offsets:  100 to  107           Dba:0x01000001
------------------------------------------------------------------------
 b4160000 00000000

 <32 bytes per line>
BBED> modify /x 2d660300
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:  100 to  107           Dba:0x01000001
------------------------------------------------------------------------
 2d660300 00000000

 <32 bytes per line>
BBED> verify
DBVERIFY - 验证正在启动
FILE =C:\ORACLE\ORADATA\ORA9I\USERS02.DBF
BLOCK = 1

块 1 已毁坏
***
Corrupt block relative dba: 0x01000001 (file 0, block 1)
Bad check value found during verification
Data in bad block -
 type: 11 format: 2 rdba: 0x01000001
 last change scn: 0x0000.00000000 seq: 0x1 flg: 0x04
 consistency value in tail: 0x00000b01
 check value in block header: 0xc78e, computed block checksum: 0x715a
 spare1: 0x0, spare2: 0x0, spare3: 0x0
***
DBVERIFY - 验证完成

检查的总块数：1
已处理的总块数（数据）：0
无法处理的总块数（数据）：0
```

```
已处理的总块数（索引）：0
无法处理的总块数（索引）：0
空的总块数：0
标记为损坏的总数块：1
汇入的块总数：0
BBED> sum apply
Check value for File 4, Block 1:
current = 0xb6d4, required = 0xb6d4
```

此时如果尝试恢复数据文件，则会遇到如下错误，这是提示创建时间错误，通过 file$仍然可以获得这个信息，需要同样修改。

```
SQL> recover datafile 4;
ORA-00283: recovery session canceled due to errors
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
ORA-01122: database file 4 failed verification check
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
ORA-01202: wrong incarnation of this file - wrong creation time
```

创建时间存储于文件头偏移量 108 位置：

```
   ub4 kcvfhcrt                            @108      0x2e086327
   ub4 kcvfhrlc                            @112      0x2cdbbe56
```

修改数据文件的创建时间信息：

```
BBED> set offset 108
        OFFSET          108
BBED> dump
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
 Block: 1                Offsets:  108 to  115           Dba:0x01000001
------------------------------------------------------------------------
 72bedb2c 56bedb2c

 <32 bytes per line>
BBED> modify /x 2763082e
 File: C:\ORACLE\ORADATA\ORA9I\USERS02.DBF (4)
```

```
 Block: 1                Offsets:  108 to  115           Dba:0x01000001
------------------------------------------------------------------------
 2763082e 56bedb2c

 <32 bytes per line>
```

BBED> sum apply

```
Check value for File 4, Block 1:
current = 0x6952, required = 0x6952
```

旧的控制文件与新的数据文件

此时执行恢复，数据库提示控制文件比数据文件旧，我们需要重建控制文件：

SQL> recover datafile 4;

```
ORA-00283: recovery session canceled due to errors
ORA-01122: database file 4 failed verification check
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
ORA-01207: file is more recent than controlfile - old controlfile
```

通过如下步骤重建控制文件（正确的重建控制文件是每个 DBA 应当具备的基本功）：

SQL> alter database backup controlfile to trace;

```
Database altered.
```

SQL> shutdown immediate;

```
Database closed.
Database dismounted.
ORACLE instance shut down.
ORACLE instance started.

Total System Global Area  126950956 bytes
Fixed Size                   454188 bytes
Variable Size              92274688 bytes
Database Buffers           33554432 bytes
Redo Buffers                 667648 bytes
```

SQL> CREATE CONTROLFILE REUSE DATABASE "ORA9I" NORESETLOGS ARCHIVELOG

```
  2  --   SET STANDBY TO MAXIMIZE PERFORMANCE
  3       MAXLOGFILES 5
  4       MAXLOGMEMBERS 3
  5       MAXDATAFILES 100
  6       MAXINSTANCES 1
  7       MAXLOGHISTORY 226
  8  LOGFILE
  9    GROUP 1 'C:\ORACLE\ORADATA\ORA9I\REDO01.LOG'   SIZE 10M,
 10    GROUP 2 'C:\ORACLE\ORADATA\ORA9I\REDO02.LOG'   SIZE 10M,
 11    GROUP 3 'C:\ORACLE\ORADATA\ORA9I\REDO03.LOG'   SIZE 10M
 12  -- STANDBY LOGFILE
 13  DATAFILE
 14    'C:\ORACLE\ORADATA\ORA9I\SYSTEM01.DBF',
 15    'C:\ORACLE\ORADATA\ORA9I\UNDOTBS01.DBF',
 16    'C:\ORACLE\ORADATA\ORA9I\USERS01.DBF',
 17    'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
 18  CHARACTER SET ZHS16GBK
 19  ;
Control file created.
```

此时尝试打开数据库，提示文件 4 需要恢复：

```
SQL> alter database open;
alter database open
*
ERROR at line 1:
ORA-01113: file 4 needs media recovery
ORA-01110: data file 4: 'C:\ORACLE\ORADATA\ORA9I\USERS02.DBF'
```

执行恢复，将从新的日志读取信息，文件 4 得以成功恢复：

```
SQL> recover datafile 4;
Media recovery complete.
SQL> alter database open;
Database altered.
```