AdvFS in-memory structures

Module 3

Copyright (C) 2008 Hewlett-Packard Development Company, L.P.

Objectives

- **Define VFS structures**
- **Describe FAS layer in-memory structures**
- **Describe BAS layer in-memory structures**
- **Explain other in-memory structures**

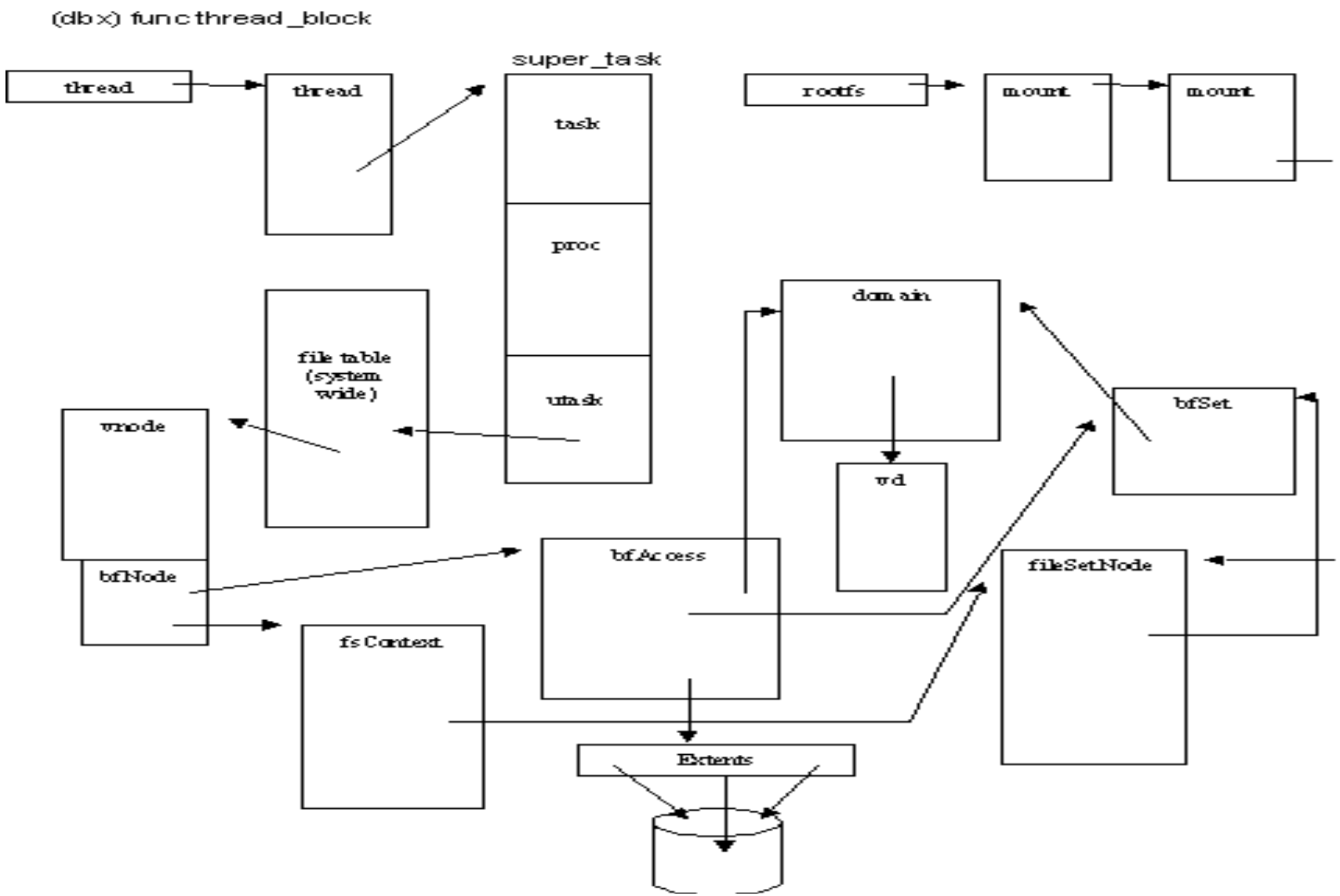
Copyright (C) 2008 Hewlett-Packard Development Company,

Overview of in-memory structures

- **VFS layer**
  - vnode and mount structures
- **FAS layer**
  - POSIX file and fileset structures
- **BAS layer**
  - bitfile and bitfile-set structures

Copyright (C) 2008 Hewlett-Packard Development Company,

Big picture of data structure linkage



Copyright (C) 2008 Hewlett-Packard Development Company,

VFS-specific structures

File descriptor:

- Returned from the open(2) system call
- Used in the per-process `utask` structure
- Points (indirectly) to file structure

Copyright (C) 2008 Hewlett-Packard Development Company,

Open system call returning a file descriptor

```
int main(void)
{
  int fd, uid, pid, bytesread;
  fd = open("/usr/bruden/ob_1", O_RDWR | O_CREAT,
O777);
  if (fd == -1)
  {
    perror("open failed ");
    exit(EXIT_FAILURE);
  }
  printf("file opened -- file descriptor is %d.\n",fd);
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Data structure for open() call

File structure for file descriptor (fd)

- Contains file credentials
- Contains file offset
- Points to vnode

vnode has a file system-specific extension leading to FAS in-memory information

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the file structure

(dbx) whatis struct file

```
struct file {
    simple_lock_data_t  f_incore_lock;
    int  f_flag;
    uint_t  f_count;
    int  f_type;
    int  f_msgcount;
    struct ucred * f_cred;
    struct fileops * f_ops;
    caddr_t  f_data;          <== This field will most likely point to a vnode
    union  {
        off_t  fu_offset;
        struct file * fu_freef;
    }  f_u;
    uint_t  f_io_lock;
    int  f_io_waiters;
};
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Using utask to get open file information

```
(dbx) set $pid=953
(dbx) p (*(struct super_task *)thread.task).utask
struct {
    uu_comm = "openone"
    uu_maxuprc = 64
    uu_logname = 0xfffffc0002fcc4a0 = "root"
(...)
    uu_file_state = struct {
(...)
        uf_entry = {
            [0] 0xfffffc00017bd700   <== Indirectly points
            [1] (nil)                    to  file structure
            [2] (nil)
            [3] (nil)
(...)
      }
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Getting to the vnode

```
(dbx) p *(struct vnode *)(*(struct super_task
*)thread.task)
     .utask.uu_file_state.uf_entry[0][3].ufe_ofile.f_data
struct {
    v_lock = 0
(...)
    v_type = VREG                          <== Regular file
    v_tag = VT_MSFS                        <== AdvFS
    v_mount = 0xfffffc0005ab2a80           <== To mount structure
    v_mountedhere = (nil)
    v_op = 0xfffffc00006b01d0
    v_freef = (nil)
    v_freeb = (nil)
    v_mountf = 0xfffffc000367bb00
    v_mountb = 0xfffffc0001c3a958
       (...)
    v_data = "^"       <== Begins file system specific information
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

vnode structure

- One per opened file
- Points to mount structure
- Points to VM object
- Points to vnode switch table
- Ends with a file system specific private area extension

Copyright (C) 2008 Hewlett-Packard Development Company,

mount structure

- One per active file system
- Points to root vnode
- Start of linked list of file system vnodes
- Points to VFS switch table
- Points to file system specific private area

Copyright (C) 2008 Hewlett-Packard Development Company,

Viewing a mount structure using rootfs (1 of 3)

```
(dbx) p *rootfs.m_nxt.m_nxt
struct {
    m_lock = 18446739675758144512
    m_flag = 20480
    m_funnel = 0
    m_nxt = 0xfffffc0005ab2d80   <==  To next mount structure
    m_prev = 0xfffffc0005ab3380
    m_op = 0xfffffc00006af990
    m_vnodecovered = 0xfffffc0001a8f200
    m_mounth = 0xfffffc0004d5a240
    m_vlist_lock = 0
    m_exroot = 0
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Viewing a mount structure using rootfs (2 of 3)

```
    m_uid = 0
    m_stat = struct {
        f_type = 10
        f_flags = 20480
        f_fsize = 512
        f_bsize = 8192
        f_blocks = 1426112
        f_bfree = 400688
        f_bavail = 324960
        f_files = 582215
        f_ffree = 558528
(...)
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Viewing a mount structure using rootfs (3 of 3)

```
        f_mntonname = 0xfffffc0000d34c20 = "/usr"
        f_mntfromname = 0xfffffc0000d34940 =
"usr_domain#usr"
   (...)
            msfs_args = struct {
               id = struct {
                   id1 = 937059922
                   id2 = 653520
                   tag = 1
               }
            }
(...)
```

Copyright (C) 2008 Hewlett-Packard Development Company,

FAS layer structures

- **`bfnode` structure**
  - AdvFS vnode
  - Points to BAS layer `bsAccess` structure
- **Fileset context**
  - Points to parent fileset
  - `fsContext` structure
- **Fileset node**
  - AdvFS private mount information
  - `fileSetNode` structure

Copyright (C) 2008 Hewlett-Packard Development Company,

In-memory per file structures

File access subsystem provides:

- Interface to storage system, the bitfile access subsystem
- Per-file statistics and directories
- Access to symbolic links stored in bitfile metadata table entries

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the bfNode structure

```
/*
 * bfNode is the msfs structure at the end of a vnode
 */

typedef struct bfNode {
        struct bfAccess *accessp;
        struct fsContext *fsContextp;
        bfTagT tag;
        bfSetIdT bfSetId;
} bfNodeT;
```

- **Source location: `msfs/ms_osf.h`**

`(dbx) set $bf=(struct bfNode *)&($vn->v_data)`

Copyright (C) 2008 Hewlett-Packard Development Company,

Accessing bfNode structure using an alias (1 of 2)

```
(dbx) alias v5_get_ofile_bfNode_struct(pidd,fd)
"set $pid=pidd; p *(struct bfNode *)&((struct vnode
*)(*(struct super_task
*)thread.task).utask.uu_file_state.uf_entry[0][fd].ufe_
ofile.f_data).v_data"

(dbx)v5_get_ofile_bfNode_struct(953,3)
struct {
    accessp = 0xfffffc0004d94d88
    fsContextp = 0xfffffc0004d5ae70
    tag = struct {
        num = 23704
        seq = 32770
    }
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Accessing bfNode structure using an alias (2 of 2)

```
    bfSetId = struct {
        domainId = struct {
            tv_sec = 937059922
            tv_usec = 653520
        }
        dirTag = struct {
            num = 1
            seq = 32769
        }
    }
}
(dbx) set $bfaccess=0xfffffc0004d94d88
(dbx) set $fscontext=0xfffffc0004d5ae70
```

Copyright (C) 2008 Hewlett-Packard Development Company,

fsContext structure

- **Located through the bfNode structure**
- **UNIX (POSIX) information about a file**
  - Rather than the bitfile
- **Contains:**
  - Quota information
  - Tag of fileset
  - Tag of file’s parent directory
  - File stats

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the fsContext structure (1 of 3)

```
struct fsContext {
    short initialized;          /* zero if fsContext is
                                   not initialized */
    short quotaInitialized;     /* zero if quota stuff is
                                   not initialized */
    bfTagT undel_dir_tag;       /* tag of undelete
                                   directory */
    long fs_flag;               /* flag word - see below */
    int dirty_stats;            /* flag for directories,
                                   says update the stats in
                                   the parent directory entry */
    int dirty_alloc;            /* set if stats from
                                   an allocating write
                                   are not on disk (ICHGMETA)
*/
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the fsContext structure (2 of 3)

```
    lock_data_t file_lock;          /* Use an OSF complex
                                       lock (read_write_lock) */
    long dirstamp;                  /* stamp to determine
                                       directory changes */
    mutexT fsContext_mutex;         /* mutex to take out locks
                                       on this structure */
#ifdef ADVFS_DEBUG
    char file_name[30];             /* first 29 chars of file
                                         name */
#endif
    bfTagT bf_tag;                  /* the tag for the file */
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the fsContext structure (3 of 3)

```
    long last_offset;              /* the offset of the
                                      last found entry */
    struct fs_stat dir_stats;     /* stats */
    struct fileSetNode *fileSetNode;  /* pointer to
                                      per-fileset info */
    struct dQuot *diskQuot[MAXQUOTAS]; /* pointers to
                                      quota structs */

};
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the fsContext structure (1 of 3)

```
(dbx)p *(struct fsContext *)$fscontext
struct {
    initialized = 1
    quotaInitialized = 1
    undel_dir_tag = struct {
        num = 0
        seq = 0
    }
    fs_flag = 0
(...)
    bf_tag = struct {
        num = 23704
        seq = 32770
    }
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the fsContext structure (2 of 3)

```
    last_offset = 0
    dir_stats = struct {
        st_ino = struct {
            num = 23704
            seq = 32770
        }
        st_mode = 33261
        st_uid = 0
        st_gid = 0
        st_rdev = 0
        st_size = 31114
(...)
        fragId = struct {
            frag = 58113
```

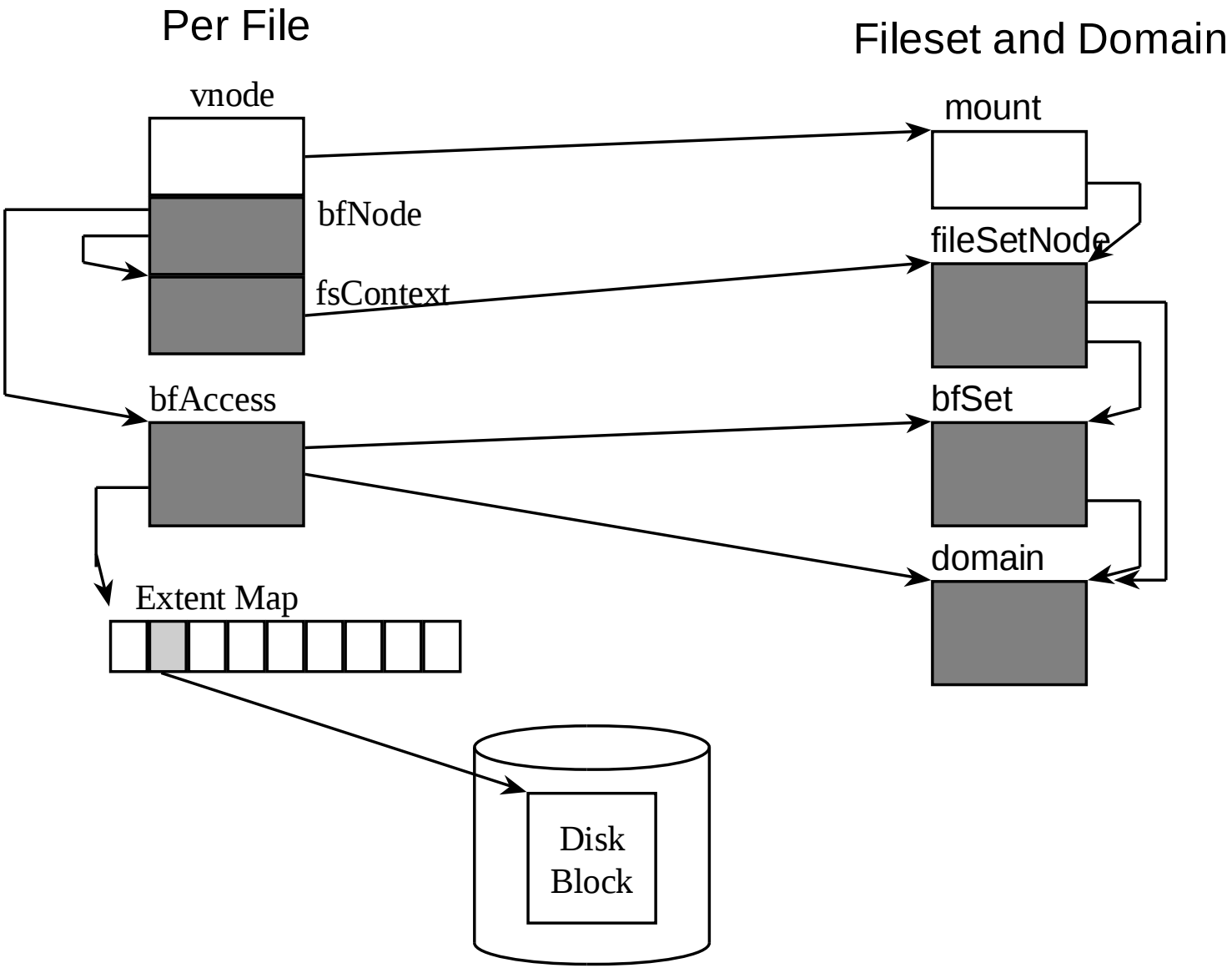
Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the fsContext structure (3 of 3)

```
            type = BF_FRAG_7K
        }
        st_nlink = 1
        st_unused_1 = 0
        fragPageOffset = 3
        st_unused_2 = 0
    }
    fileSetNode = 0xfffffc0005ab5088
    diskQuot = {
        [0] 0xfffffc0005ac7088
        [1] 0xfffffc0005ac7148
    }
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

In-memory per file structures



Copyright (C) 2008 Hewlett-Packard Development Company,

In-memory per fileset structures

- The mount structure for this file system is linked to the mount structure of the root file system (`m_nxt`) which is found through the global symbol rootfs
- The mounted file system's mount structure points (`m_data`) to the file system specific mount structure, in this case the AdvFS `fileSetNode` structure
- The vnode of the mounted-upon directory is set to point (`v_mountedhere`) to the mounted file system's mount structure to represent where the file system has been mounted
- This mount structure points (`m_vnodecovered`) back to the vnode
- Attached to the vnodes of the active files of the mounted file system are `bfNode` data structures

Copyright (C) 2008 Hewlett-Packard Development Company,

fileSetNode structure

- **AdvFS specific mount structure**
- **Includes pointers to:**
  - domain structure
  - vnode for root
  - mount structure

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the fileSetNode structure (1 of 3)

```
typedef struct fileSetNode {
    struct fileSetNode *fsNext;
    struct fileSetNode **fsPrev;
    bfTagT rootTag;             /* tag of root
                                   directory */
    bfTagT tagsTag;             /* tag of ".tags */
    uint_t filesetMagic;        /* magic number:
                                    structure validation */
    domainT *dmnP;
    bfAccessT *rootAccessp;     /* Access structure
                                   pointer for root */
    bfSetIdT bfSetId;
    bfSetT *bfSetp;             /* bitfile-set
                                    descriptor pointer */
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the fileSetNode structure (2 of 3)

```
struct vnode *root_vp;
int fsFlags;                    /* flags, see below */
struct mount *mountp;           /* mount table pointer */
unsigned quotaStatus;           /* see definitions
                                   below */
long blkHLimit;                 /* maximum quota
                                   blocks in fileset */
long blkSLimit;                 /* soft limit for
                                   fileset blks */
long fileHLimit;                /* maximum number of
                                   files in fileset */
long fileSLimit;                /* soft limit for
                                   fileset files */
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields of the fileSetNode structure (3 of 3)

```
    long blksUsed;              /* number of quota
                                   blocks used */
    long filesUsed;             /* number of bitfiles
                                   used */
    time_t blkTLimit;           /* time limit for
                                excessive disk blk use */
    time_t fileTLimit;          /* time limit for
                                   excessive file use */
    mutexT filesetMutex;        /* protect next two
                                   fields */
    quotaInfoT qi[MAXQUOTAS];
    fileSetStatsT fileSetStats;
} fileSetNodeT;
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the fileSetNode structure (1 of 4)

```
(dbx) p *(*(struct fsContext *)$fscontext).fileSetNode
struct {
    fsNext = (nil)
    fsPrev = 0xfffffc0005ab5348
    rootTag = struct {
        num = 2
        seq = 32769
    }
    tagsTag = struct {
        num = 3
        seq = 32769
    }
    filesetMagic = 2918187013        <== 0xadf00005
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the fileSetNode structure (2 of 4)

```
dmnP = 0xfffffc0000f24008
rootAccessp = 0xfffffc0005af7688
bfSetId = struct {
    domainId = struct {
        tv_sec = 937059922
        tv_usec = 653520
    }
    dirTag = struct {
        num = 1
        seq = 32769
    }
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the fileSetNode structure (3 of 4)

```
    bfSetp = 0xfffffc0005b7ca08
    root_vp = 0xfffffc0005ac98c0
    fsFlags = 0
    mountp = 0xfffffc0005ab2a80
    quotaStatus = 1421
    blkHLimit = 0
    blkSLimit = 0
    fileHLimit = 0
    fileSLimit = 0
    blksUsed = 1025520
    filesUsed = 23689
(...)
```

Copyright (C) 2008 Hewlett-Packard Development Company,

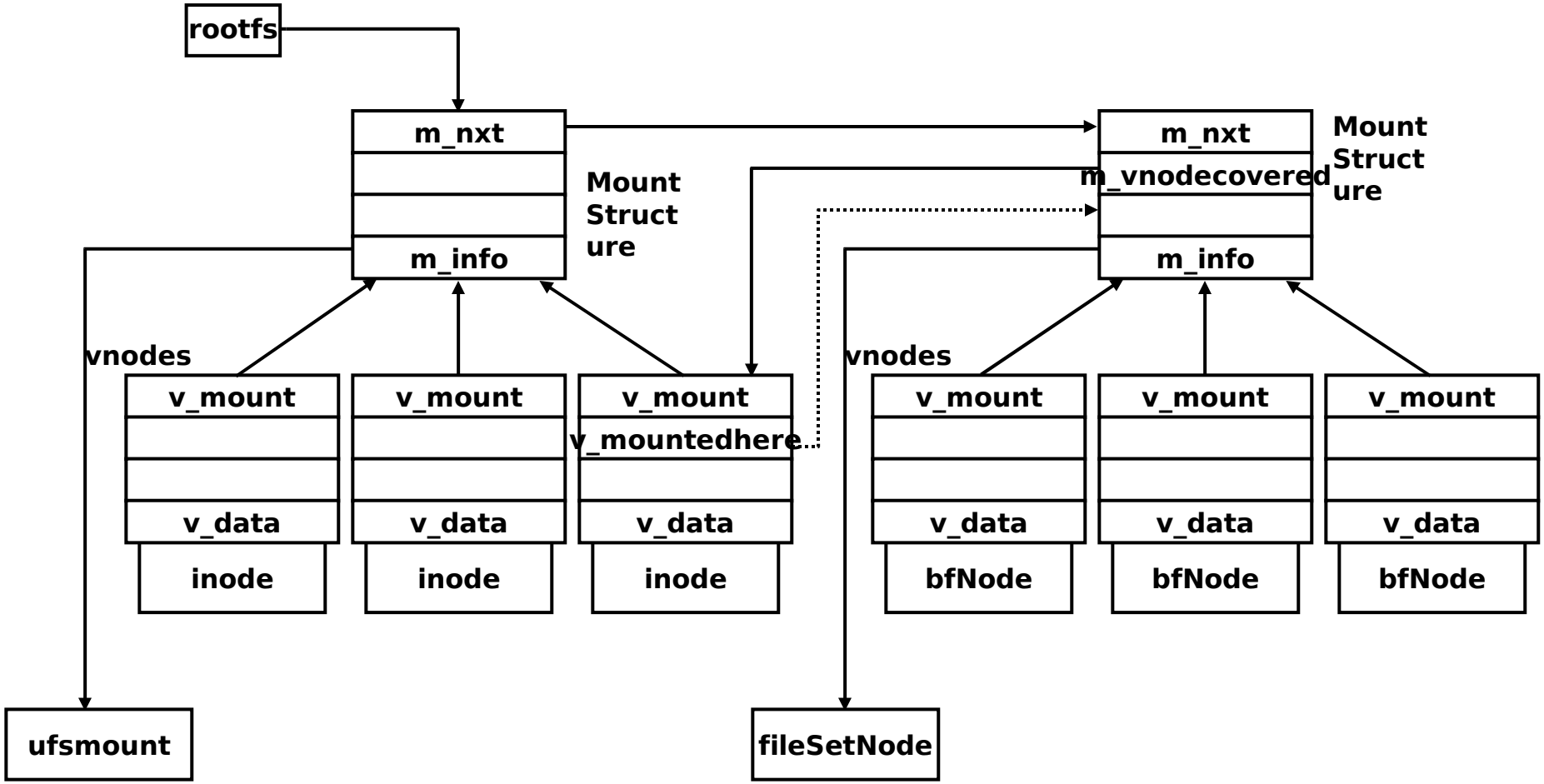
Displaying the fileSetNode structure (4 of 4)

```
      fileSetStats = struct {
         msfs_lookup = 19671
         lookup = struct {
             hit = 17494
             hit_not_found = 1079
             miss = 1098

         }
         msfs_create = 2
         msfs_mknod = 0
(...)
            }
}
(dbx) set $bfset=0xfffffc0005b7ca08
(dbx) set $domain=0xfffffc0000f24008
```

Copyright (C) 2008 Hewlett-Packard Development Company,

In-memory per-fileset structures



Copyright (C) 2008 Hewlett-Packard Development Company,

Fileset quota structures

- **fileSetNode contains an array of `quotaInfo` structures**
  - Identify fileset quotas
- **fileSetNode contains several limit fields**
  - For fields set by `chfsets`

Copyright (C) 2008 Hewlett-Packard Development Company,

User and group quota structures

- **fsContext points to `dQuot` structures**
- **Kernel maintains disk quota cache**
  - Table is `DqHashTbl`
  - Access via `dqget()`
  - `msfs/msfs/fs_quota.h` contains includes
  - `msfs/fs/fs_quota.c` contains routines

Copyright (C) 2008 Hewlett-Packard Development Company,

BAS layer structure overview

- **`bfAccess` structure**
  - bitfile structure
  - One per open file
- **`bfSet`**
  - bitfile-set structure
  - One per fileset
- **`domainT`**
  - File domain
  - One per domain
- **`vd` structure**
  - Virtual disks
  - One per AdvFS volume

Copyright (C) 2008 Hewlett-Packard Development Company,

Access to BAS structures

- **In previous releases**
  - BAS structures were accessed through handles
- **In current release**
  - BAS structures are accessed through pointers

Copyright (C) 2008 Hewlett-Packard Development Company,

bfAccess structure

- **In-memory state of a bitfile**
- **Contains:**
  - Links to other bitfile access structures
  - Pointer to a vnode
  - Pointer to a vm object
  - Highest LSN written to a log
  - If a clone - pointers to next clone's `bfAccess` structure
  - Bitfile set pointer
  - Domain pointer
  - Primary metadata cell ID
  - Volume containing primary mcell

Copyright (C) 2008 Hewlett-Packard Development Company,

Managing bfAccess structures

- **Allocated as needed**
- **Free Access list**
  - Linked list of available structures
- **Closed Access list**
  - Closed and dirty bitfiles that need a bit more work before freeing

Copyright (C) 2008 Hewlett-Packard Development Company,

Accessing extent data through bfAccess (1 of 2)

```
(dbx) p *(*(struct bfAccess *)$bfaccess).xtnts.xtntMap.subXtntMap[0].bsXA[0]
struct {
    bsPage = 0
    vdBlk = 222416
}
(dbx) q
#
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Accessing extent data through bfAccess (2 of 2)

```
# showfile -x ob_1
         Id  Vol  PgSz  Pages  XtntType  Segs  SegSz  I/O
Perf  File
  5c98.8002    1    16      3    simple    **     **  async
100%  ob_1
    extentMap: 1
        pageOff    pageCnt      vol    volBlock    blockCnt
              0          3        1      222416          48
        extentCnt: 1
#
```

Copyright (C) 2008 Hewlett-Packard Development Company,

bfSet structure

- **Lower-level BAS structure for a fileset**
- **Includes:**
  - Pointer to the fileset's domain
  - Tag and references to this set's tag directory
  - Cloned/master state
  - Fragment file information
  - Back pointer to fileset node

Copyright (C) 2008 Hewlett-Packard Development Company,

Finding bfSet structures

`BfSetHashTbl`

- **Elements indirectly point to `bfSet` structures**

Copyright (C) 2008 Hewlett-Packard Development Company,

Hash key for BfSetHashTbl

```
#define BFSET_GET_HASH_KEY( _bfSetId )

 ( (_bfSetId).domainId.tv_sec +
(_bfSetId).dirTag.num )
```

Copyright (C) 2008 Hewlett-Packard Development Company,

domain structure

- **One per domain**
- **Includes:**
  - root tag directory references
  - log location and state information
  - overall buffering state information

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (1 of 10)

```
(dbx) p *(domainT *)$domain
struct {
    mutex = struct {
        mutex = 0
    }
    dmnMagic = 2918187011
    dmnFwd = 0xfffffc0000f24008
    dmnBwd = 0xfffffc0000f24008
    dmnHashlinks = struct {
        dh_links = struct {
            dh_next = 0xfffffc0000f24008
            dh_prev = 0xfffffc0000f24008
        }
        dh_key = 937059922
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (2 of 10)

```
}
dmnVersion = 4
state = BFD_ACTIVATED

domainId = struct {
    tv_sec = 937059922
    tv_usec = 653520
}
dualMountId = struct {
    tv_sec = 0
    tv_usec = 0
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (3 of 10)

```
bfDmnMntId = struct {
    tv_sec = 937392621
    tv_usec = 874423
}
dmnAccCnt = 4
dmnRefWaiters = 0
activateCnt = 2
mountCnt = 2
bfSetDirp = 0xfffffc0005b7c788
bfSetDirTag = struct {
    num = 4294967288
    seq = 0
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (4 of 10)

```
    (...)
      bfSetHead = struct {
        bfsQfwd = 0xfffffc0005b7cce8
        bfsQbck = 0xfffffc0005b7c7e8
    }
    bfSetDirAccp = 0xfffffc0005af8488
    ftxLogTag = struct {
        num = 4294967287
        seq = 0
    }
    ftxLogP = 0xfffffc0005baec48
    ftxLogPgs = 512
    logAccessp = 0xfffffc0005af8908
(...)
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (5 of 10)

```
        domainName = "usr_domain"
    majorNum = 2055
    flag = BFD_NORMAL
    lsnLock = struct {
        mutex = 0
    }
    lsnList = struct {
        lsnFwd = 0xfffffe04075c0e68
        lsnBwd = 0xfffffe04075c0e68
(...)
        vdCnt = 1
    vdpTbl = {
        [0] 0xfffffc0000f2b508
        [1] (nil)
        [2] (nil)
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (6 of 10)

```
(...)
    bcStat = struct {
        pinHit = 14402
        pinHitWait = 784
        pinRead = 0
        refHit = 24821
        refHitWait = 69
        raBuf = 2458
        ubcHit = 1418
        unpinCnt = struct {
            lazy = 14162
            blocking = 66
            clean = 11
            log = 2172
        }
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (7 of 10)

```
        derefCnt = 28516
        devRead = 3025
        devWrite = 3229
(...)
    bmtStat = struct {
        fStatRead = 0
        fStatWrite = 5316
        resv1 = 0
        resv2 = 0
        bmtRecRead = {
            [0] 0
            [1] 0
(...)
            [21] 0
        }
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (8 of 10)

```
        bmtRecWrite = {
            [0] 0
            [1] 0
            [2] 97
            [3] 0
(...)
            [21] 0
        }
    }
    logStat = struct {
        logWrites = 313
        transactions = 9860
        segmentedRecs = 3
        logTrims = 0
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (9 of 10)

```
        wastedWords = 27558
        maxLogPgs = 102
        minLogPgs = 0
        maxFtxWords = 2127
        maxFtxAgent = 91
        maxFtxTblSlots = 15
        oldFtxTblAgent = 34
        excSlotWaits = 0
        fullSlotWaits = 0
        rsv1 = 0
        rsv2 = 0
        rsv3 = 0
        rsv4 = 0
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Displaying the domain structure (10 of 10)

```
    totalBlks = 1426112
    freeBlks = 324480
    dmn_panic = 0
(...)
    smsync_policy = 0
    metaPagep = 0xfffffe0400299008
    fs_full_time = 0
}
```

Copyright (C) 2008 Hewlett-Packard Development Company,

vd structure

- **Per virtual disk structure**
- **Includes:**
  - Pointer to device vnode
  - Pointer to RBMT, BMT, and SBM bitfiles
  - Physical characteristics of device
  - I/O queuing information

Copyright (C) 2008 Hewlett-Packard Development Company,

Free space cache

Characteristics of the free space cache include:

- **Per volume in-core structure**
  - struct `stgDesc`
- **Linked list of contiguous free clusters**
- **Each entry gives the starting block and size of each free area**

Copyright (C) 2008 Hewlett-Packard Development Company,

Fields in stgDesc structure

```
/*
 * stgDescT - Describes a contiguous set of
 * available (free) vd blocks
    . . .
*/
typedef struct stgDesc {
    uint32T start_clust;    /* vd cluster number
                            of first free cluster */
    uint32T num_clust;      /* number of free
                               clusters */
    struct stgDesc *prevp;
    struct stgDesc *nextp;
} stgDescT;
```

Copyright (C) 2008 Hewlett-Packard Development Company,

Bitfile buffer descriptor

- **Descriptor for bitfile pages**
- **struct `bsBuf`**
  - Lots of fields for doubly linked lists
  - Log record addresses
  - Page address (Domain, bitfile-set, fileset, page)
  - Physical location
  - bfAccess structure
  - I/O descriptor information and queues (Migrating pages may have more than one I/O descriptor)

Copyright (C) 2008 Hewlett-Packard Development Company,

I/O descriptor

- **Links for the I/O queues**
- **Block descriptor**
  - Virtual disk
  - Block
- **Address of buffer**
- **Pointer to bsBuf structure**

Copyright (C) 2008 Hewlett-Packard Development Company,

FTX state structure

- **struct `ftx` or `ftxStateT`**
- **Fields include:**
  - Log record numbers
    - First and last written
    - Undo back link

Copyright (C) 2008 Hewlett-Packard Development Company,

Learning check

Lab 3


hp
invent