```
**********************************************************
    1560 Tue Jun 11 09:10:19 2019
new/usr/src/cmd/smbsrv/test-msgbuf/test_main.c
11227 smb code needs smatch fixes
**********************************************************
```

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2018 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2019 Joyent, Inc.
 */

/*
 * Test & debug program for smb_msgbuf.c and smb_mbuf_marshaling.c
 */

#include <sys/types.h>
#include <sys/debug.h>

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <unistd.h>

#include "test_defs.h"


int
main(int argc, char *argv[])
{

        test_conv();
        test_mbmarshal();
        test_msgbuf();

        return (0);
}

void
hexdump(const uchar_t *buf, int len)
{
        int idx;
        char ascii[24];
        char *pa = ascii;

        memset(ascii, '\0', sizeof (ascii));

        idx = 0;
        while (len--) {
                if ((idx & 15) == 0) {
                        printf("%04X: ", idx);
                        pa = ascii;
                }
                if (*buf > ' ' && *buf <= '~')
                        *pa++ = *buf;
                else
```



```
                        *pa++ = '.';
                printf("%02x ", *buf++);

                idx++;
                if ((idx & 3) == 0) {
                        *pa++ = ' ';
                        (void) putchar(' ');
                        putchar(' ');
                }
                if ((idx & 15) == 0) {
                        *pa = '\0';
                        printf("%s\n", ascii);
                }
        }

        if ((idx & 15) != 0) {
                *pa = '\0';
                /* column align the last ascii row */
                while ((idx & 15) != 0) {
                        if ((idx & 3) == 0)
                                (void) putchar(' ');
                                putchar(' ');
                        printf("   ");
                        idx++;
                }
                printf("%s\n", ascii);
        }
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   13679 Tue Jun 11 09:10:20 2019
new/usr/src/cmd/smbsrv/testoplock/tol_main.c
11227 smb code needs smatch fixes
**********************************************************
```

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2018 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2019 Joyent, Inc.
 */

/*
 * Test & debug program for oplocks
 *
 * This implements a simple command reader which accepts
 * commands to simulate oplock events, and prints the
 * state changes and actions that would happen after
 * each event.
 */

#include <sys/types.h>
#include <sys/debug.h>
#include <sys/stddef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <unistd.h>

#include <smbsrv/smb_kproto.h>
#include <smbsrv/smb_oplock.h>

#define OPLOCK_CACHE_RWH        (READ_CACHING | HANDLE_CACHING | WRITE_CACHING)
#define OPLOCK_TYPE     (LEVEL_TWO_OPLOCK | LEVEL_ONE_OPLOCK |\
                        BATCH_OPLOCK | OPLOCK_LEVEL_GRANULAR)

#define MAXFID 10

smb_node_t root_node, test_node;
smb_ofile_t  ofile_array[MAXFID];
smb_request_t test_sr;
uint32_t last_ind_break_level;
char cmdbuf[100];

extern const char *xlate_nt_status(uint32_t);

#define BIT_DEF(name) { name, #name }

struct bit_defs {
        uint32_t mask;
        const char *name;
} state_bits[] = {
```

*unchanged_portion_omitted_*

```
static void
do_open(int fid, char *arg2)
```



```
{
        smb_node_t *node = &test_node;
        smb_ofile_t *ofile = &ofile_array[fid];

        /*
         * Simulate an open (minimal init)
         */
        if (ofile->f_refcnt) {
                printf("open fid %d already opened\n");
                return;
        }

        if (arg2 != NULL) {
                (void) strlcpy((char *)ofile->TargetOplockKey, arg2,
        if (arg2 != NULL)
                strlcpy((char *)ofile->TargetOplockKey, arg2,
                    SMB_LEASE_KEY_SZ);
        }

        ofile->f_refcnt++;
        node->n_open_count++;
        smb_llist_insert_tail(&node->n_ofile_list, ofile);
        printf(" open %d OK\n", fid);
}
```

*unchanged_portion_omitted_*

```
int
main(int argc, char *argv[])
{
        smb_node_t *node = &test_node;
        char *cmd;
        char *arg1;
        char *arg2;
        char *savep;
        char *sep = " \t\n";
        char *prompt = NULL;
        int fid;

        if (isatty(0))
                prompt = "> ";

        smb_llist_constructor(&node->n_ofile_list, sizeof (smb_ofile_t),
            offsetof(smb_ofile_t, f_node_lnd));

        for (fid = 0; fid < MAXFID; fid++) {
                smb_ofile_t *f = &ofile_array[fid];

                f->f_magic = SMB_OFILE_MAGIC;
                mutex_init(&f->f_mutex, NULL, MUTEX_DEFAULT, NULL);
                f->f_fid = fid;
                f->f_ftype = SMB_FTYPE_DISK;
                f->f_node = &test_node;
        }

        for (;;) {
                if (prompt) {
                        (void) fputs(prompt, stdout);
                        fputs(prompt, stdout);
                        fflush(stdout);
                }

                cmd = fgets(cmdbuf, sizeof (cmdbuf), stdin);
                if (cmd == NULL)
                        break;
                if (cmd[0] == '#')
                        continue;
```

```
412                 if (prompt == NULL) {
413                         /* Put commands in the output too. */
414                         (void) fputs(cmdbuf, stdout);
412                         fputs(cmdbuf, stdout);
415                 }
416                 cmd = strtok_r(cmd, sep, &savep);
417                 if (cmd == NULL)
418                         continue;

420                 /*
421                  * Commands with no args
422                  */
423                 if (0 == strcmp(cmd, "help")) {
424                         (void) fputs(helpstr, stdout);
422                         fputs(helpstr, stdout);
425                         continue;
426                 }

428                 if (0 == strcmp(cmd, "show")) {
429                         do_show();
430                         continue;
431                 }

433                 /*
434                  * Commands with one arg (the FID)
435                  */
436                 arg1 = strtok_r(NULL, sep, &savep);
437                 if (arg1 == NULL) {
438                         fprintf(stderr, "%s missing arg1\n", cmd);
439                         continue;
440                 }
441                 fid = atoi(arg1);
442                 if (fid <= 0 || fid >= MAXFID) {
443                         fprintf(stderr, "%s bad FID %d\n", cmd, fid);
444                         continue;
445                 }

447                 if (0 == strcmp(cmd, "close")) {
448                         do_close(fid);
449                         continue;
450                 }
451                 if (0 == strcmp(cmd, "brk-parent")) {
452                         do_brk_parent(fid);
453                         continue;
454                 }
455                 if (0 == strcmp(cmd, "brk-handle")) {
456                         do_brk_handle(fid);
457                         continue;
458                 }
459                 if (0 == strcmp(cmd, "brk-read")) {
460                         do_brk_read(fid);
461                         continue;
462                 }
463                 if (0 == strcmp(cmd, "brk-write")) {
464                         do_brk_write(fid);
465                         continue;
466                 }

468                 /*
469                  * Commands with an (optional) arg2.
470                  */
471                 arg2 = strtok_r(NULL, sep, &savep);

473                 if (0 == strcmp(cmd, "open")) {
474                         do_open(fid, arg2);
```



```
475                         continue;
476                 }
477                 if (0 == strcmp(cmd, "req")) {
478                         do_req(fid, arg2);
479                         continue;
480                 }
481                 if (0 == strcmp(cmd, "ack")) {
482                         do_ack(fid, arg2);
483                         continue;
484                 }
485                 if (0 == strcmp(cmd, "brk-open")) {
486                         do_brk_open(fid, arg2);
487                         continue;
488                 }
489                 if (0 == strcmp(cmd, "brk-setinfo")) {
490                         do_brk_setinfo(fid, arg2);
491                         continue;
492                 }
493                 if (0 == strcmp(cmd, "move")) {
494                         do_move(fid, arg2);
495                         continue;
496                 }
497                 if (0 == strcmp(cmd, "waiters")) {
498                         do_waiters(fid, arg2);
499                         continue;
500                 }

502                 fprintf(stderr, "%s unknown command. Try help\n", cmd);
503         }
504         return (0);
505 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7373 Tue Jun 11 09:10:20 2019
new/usr/src/uts/common/fs/smbsrv/smb_alloc.c
11227 smb code needs smatch fixes
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*
 * Copyright 2019 Joyent, Inc.
 */

#include <sys/types.h>
#include <sys/sunddi.h>
#include <sys/kmem.h>
#include <sys/sysmacros.h>
#include <smbsrv/smb_kproto.h>
#include <smbsrv/alloc.h>

#define SMB_SMH_MAGIC           0x534D485F      /* 'SMH_' */
#define SMB_SMH_VALID(_smh_)    ASSERT((_smh_)->smh_magic == SMB_SMH_MAGIC)
#define SMB_MEM2SMH(_mem_)      ((smb_mem_header_t *)(_mem_) - 1)

typedef struct smb_mem_header {
        uint32_t        smh_magic;
        size_t          smh_size;
        smb_request_t   *smh_sr;
        list_node_t     smh_lnd;
} smb_mem_header_t;
```

*unchanged_portion_omitted_*

```
/*
 * Allocate or resize memory previously allocated.
 *
 * sr   If not NULL, request the memory is associated with.
 *
 * ptr  Memory address
 *
 * size New size
 *
 * zero If true zero out the extra space or the truncated space.
 */
static void *
smb_realloc(smb_request_t *sr, void *ptr, size_t size, boolean_t zero)
{
```



```
        smb_mem_header_t        *smh;
        void                    *new_ptr;

        if (ptr == NULL)
                return (smb_alloc(sr, size, zero));

        smh = SMB_MEM2SMH(ptr);
        SMB_SMH_VALID(smh);
        ASSERT(sr == smh->smh_sr);

        if (size == 0) {
                smb_free(sr, ptr, zero);
                return (NULL);
        }
        if (smh->smh_size >= size) {
                if ((zero) && (smh->smh_size > size))
                if ((zero) & (smh->smh_size > size))
                        bzero((caddr_t)ptr + size, smh->smh_size - size);
                return (ptr);
        }
        new_ptr = smb_alloc(sr, size, B_FALSE);
        bcopy(ptr, new_ptr, smh->smh_size);
        if (zero)
                bzero((caddr_t)new_ptr + smh->smh_size, size - smh->smh_size);

        smb_free(sr, ptr, zero);
        return (new_ptr);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    8990 Tue Jun 11 09:10:20 2019
new/usr/src/uts/common/smbsrv/mbuf.h
11227 smb code needs smatch fixes
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2019 Joyent, Inc.
 * Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright (c) 1982, 1986, 1988 Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the University of
 *      California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 */

#ifndef _SMBSRV_MBUF_H
```



```
#define _SMBSRV_MBUF_H

/*
 * This mbuf simulation should be replaced with (native) mblk_t support.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/kmem.h>
#include <smbsrv/string.h>

#ifdef __cplusplus
extern "C" {
#endif

#define MSIZE           256
#define MCLBYTES        8192

/*
 * Mbufs are of a single size, MSIZE (machine/machparam.h), which
 * includes overhead.  An mbuf may add a single "mbuf cluster" of size
 * MCLBYTES (also in machine/machparam.h), which has no additional overhead
 * and is used instead of the internal data area; this is done when
 * at least MINCLSIZE of data must be stored.
 */

#define MLEN            (MSIZE - sizeof (struct m_hdr)) /* normal data len */
#define MHLEN           (MLEN - sizeof (struct pkthdr)) /* data len w/pkthdr */

#define MINCLSIZE       (MHLEN + MLEN)  /* smallest amount to put in cluster */

/*
 * Macros for type conversion
 * mtod(m,t) -  convert mbuf pointer to data pointer of correct type
 */
#define mtod(m, t)      ((t)((m)->m_data))


/* header at beginning of each mbuf: */
struct m_hdr {
        struct  mbuf *mh_next;          /* next buffer in chain */
        struct  mbuf *mh_nextpkt;       /* next chain in queue/record */
        int     mh_len;                 /* amount of data in this mbuf */
        caddr_t mh_data;                /* location of data */
        short   mh_type;                /* type of data in this mbuf */
        short   mh_flags;               /* flags; see below */
};
```

_____***unchanged_portion_omitted_***

```
#define MCLGET(m, how) \
        { \
                (m)->m_ext.ext_buf = smb_mbufcl_alloc();        \
                (m)->m_data = (m)->m_ext.ext_buf;               \
                (m)->m_flags |= M_EXT;                          \
                (m)->m_ext.ext_size = MCLBYTES;                 \
                (m)->m_ext.ext_ref = smb_mbufcl_ref;            \
        }

/*
 * MFREE(struct mbuf *m, struct mbuf *nn)
 * Free a single mbuf and associated external storage.
 * Place the successor, if any, in nn.
 */
#define MFREE(m, nn) \
        { \
                if ((m)->m_flags & M_EXT) {                             \
```

```
237                         (void) (*((m)->m_ext.ext_ref))                  \
238                             ((m)->m_ext.ext_buf,                        \
236                         (*((m)->m_ext.ext_ref))((m)->m_ext.ext_buf,     \
239                             (m)->m_ext.ext_size, -1);                   \
240                         (m)->m_ext.ext_buf = 0;                         \
241                 }                                                       \
242                 (nn) = (m)->m_next;                                     \
243                 (m)->m_next = 0;                                        \
244                 smb_mbuf_free(m);                                       \
245         }



249 /*
250  * As above, for mbufs allocated with m_gethdr/MGETHDR
251  * or initialized by M_COPY_PKTHDR.
252  */
253 #define MH_ALIGN(m, len) \
254         { (m)->m_data += (MHLEN - (len)) &~ (sizeof (int32_t) - 1); }

256 #define SMB_MBC_MAGIC           0x4D42435F
257 #define SMB_MBC_VALID(p)        ASSERT((p)->mbc_magic == SMB_MBC_MAGIC)

259 typedef struct mbuf_chain {
260         uint32_t                mbc_magic;
261         volatile uint32_t       flags;          /* Various flags */
262         struct mbuf_chain       *shadow_of;     /* I'm shadowing someone */
263         mbuf_t                  *chain;         /* Start of chain */
264         int32_t                 max_bytes;      /* max # of bytes for chain */
265         int32_t                 chain_offset;   /* Current offset into chain */
266 } mbuf_chain_t;
```

*unchanged_portion_omitted_*