

new/usr/src/lib/libdisasm/common/dis_s390x.c

1

```
*****
79774 Thu Oct 15 09:59:21 2015
new/usr/src/lib/libdisasm/common/dis_s390x.c
patch fixup2
*****
_____unchanged_portion_omitted_____

2481 /*
2482  * Even if we don't know how to disassemble the instruction, we know how long
2483  * it is, so we always succeed. That is why we can get away with returning
2484  * void.
2485  */
2486 static void
2487 dis_s390(uint64_t addr, union inst *inst, char *buf, size_t buflen, int mach)
2488 {
2489     const struct inst_table *tbl = &tbl_xx[inst->raw[0]];
2490     int tmp;

2492     /* nothing to do */
2493     if (buflen == 0)
2494         return;

2496 #endif /* ! codereview */
2497     while (tbl->it_fmt == IF_TBL || tbl->it_fmt == IF_MULTI) {
2498         if (tbl->it_fmt == IF_TBL) {
2499             int idx;

2501                 idx = inst->raw[tbl->it_u.it_table.it_off];
2502                 idx >>= tbl->it_u.it_table.it_shift;
2503                 idx &= tbl->it_u.it_table.it_mask;

2505                 tbl = &tbl->it_u.it_table.it_ptr[idx];
2506             } else if (tbl->it_fmt == IF_MULTI) {
2507                 tbl = &tbl->it_u.it_multi.it_ptr[mach];
2508             }
2509         }

2511         if (tbl->it_fmt == IF_INVALID)
2512             goto inval;

2514         if ((tbl->it_u.it_inst.it_flags & mach) == 0)
2515             goto inval;

2517         tmp = snprintf(buf, buflen, "%-7s ", tbl->it_u.it_inst.it_name);
2518         if (tmp < 0)
2519             return;
2520 #endif /* ! codereview */

2522         fmt_fxns[tbl->it_fmt](addr, inst, buf + tmp, buflen - tmp,
2523             tbl->it_u.it_inst.it_flags);

2525         return;
2487         return (0);

2527 inval:
2528     (void) snprintf(buf, buflen, "??");

2492     /*
2493     * Even if we don't know how to disassemble the instruction, we know
2494     * how long it is, so we "succeed" even when we fail.
2495     */
2496     return (0);
2529 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libdisasm/common/dis_s390x.c

2

```
2543 static int
2544 dis_s390_disassemble(dis_handle_t *dhp, uint64_t addr, char *buf,
2545     size_t buflen)
2546 {
2547     union inst inst;
2548     int mach;
2549     int len;

2551     if (dhp->dh_read(dhp->dh_data, addr, &inst.raw[0], 2) != 2)
2552         return (-1);

2554     len = ILC2LEN(inst.raw[0] >> 6) - 2;

2556     if (len > 0 &&
2557         dhp->dh_read(dhp->dh_data, addr + 2, &inst.raw[2], len) != len)
2558         return (-1);

2560     switch (dhp->dh_flags & (DIS_S370 | DIS_S390_31 | DIS_S390_64)) {
2561     case DIS_S370:
2562         mach = F_370;
2563         break;
2564     case DIS_S390_31:
2565         mach = F_390;
2566         break;
2567     case DIS_S390_64:
2568         mach = F_Z;
2569         break;
2570     }

2572     dis_s390(addr, &inst, buf, buflen, mach);

2574     return (0);
2540     return (dis_s390(addr, &inst, buf, buflen, mach));
2575 }
_____unchanged_portion_omitted_____
```