

```

*****
7831 Thu Mar 20 21:16:55 2014
new/usr/src/uts/common/cpr/cpr_stat.c
3379 Typo or undefined operation in uts/common/cpr/cpr_stat.c
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright (c) 2014 Gary Mills
23  * Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  */
26 #pragma ident      "%Z%M% %I%      %E% SMI"
27 #include <sys/types.h>
28 #include <sys/ddi.h>
29 #include <sys/pte.h>
30 #include <sys/cpr.h>
31
32 /*
33  * Support routines for CPR statistic collection
34  */
35 struct cpr_event cpr_events_buf[CPR_E_MAX_EVENTNUM];
36
37 extern struct cpr_terminator cpr_term;
38
39 struct cpr_event *cpr_find_event(char *name, int new);
40
41 #define CPR_DEFAULT_PROMTIME      30
42 #define CE_START_MASK            0x8000000
43
44 /*
45  * Use ctp to specify another time point instead of the current time;
46  * Otherwise, ctp is NULL.
47  */
48 void
49 cpr_stat_event_start(char *name, cpr_time_t *ctp)
50 {
51     struct cpr_event *cep;
52     cpr_time_t tv;
53
54     if (ctp)
55         tv = *ctp;
56     else {
57         /* need relative time even when hrestime is stoped */
58         cpr_tod_get(&tv);
59     }

```

```

61     if ((cep = cpr_find_event(name, 1)) == NULL) {
62         cpr_err(CE_WARN, "cpr_stat: run out of event buffers");
63         return;
64     }
65     /*
66      * disallow entering start twice without calling end first
67      */
68     if (cep->ce_ntests & CE_START_MASK)
69         return;
70
71     cep->ce_ntests |= CE_START_MASK;
72     cep->ce_sec.stime = cep->ce_sec.etime = tv.tv_sec;
73     cep->ce_sec.ltime = 0;
74     cep->ce_sec.ltime = cep->ce_sec.ltime = 0;
75     cep->ce_msec.stime = cep->ce_msec.etime = tv.tv_nsec / 100000000;
76     cep->ce_msec.ltime = 0;
76 }

```

unchanged_portion_omitted