

```

*****
35038 Wed Jan 30 16:03:16 2019
new/usr/src/tools/cpcgen/cpcgen.c
10323 cpcgen_parse_model() gets value check wrong
*****
1 /*
2  * This file and its contents are supplied under the terms of the
3  * Common Development and Distribution License ("CDDL"), version 1.0.
4  * You may only use this file in accordance with the terms of version
5  * 1.0 of the CDDL.
6  *
7  * A full copy of the text of the CDDL should have accompanied this
8  * source. A copy of the CDDL is also available via the Internet at
9  * http://www.illumos.org/license/CDDL.
10 */

12 /*
13  * Copyright (c) 2019, Joyent, Inc.
13  * Copyright (c) 2018, Joyent, Inc.
14 */

16 /*
17  * This file transforms the perfmon data files into C files and manual pages.
18 */

20 #include <stdio.h>
21 #include <stdarg.h>
22 #include <unistd.h>
23 #include <err.h>
24 #include <libgen.h>
25 #include <libnvpair.h>
26 #include <strings.h>
27 #include <errno.h>
28 #include <limits.h>
29 #include <sys/mman.h>
30 #include <sys/param.h>
31 #include <assert.h>
32 #include <ctype.h>
33 #include <sys/types.h>
34 #include <sys/stat.h>
35 #include <fcntl.h>

37 #include <json_nvlist.h>

39 #define EXIT_USAGE      2

42 typedef struct cpc_proc {
43     struct cpc_proc *cproc_next;
44     uint_t          cproc_family;
45     uint_t          cproc_model;
46 } cpc_proc_t;
  unchanged_portion_omitted

295 /*
296  * Parse a string of the form 'GenuineIntel-6-2E' and get out the family and
297  * model.
298  */
299 static void
300 cpcgen_parse_model(char *fsr, uint_t *family, uint_t *model)
301 {
302     const char *bstr = "GenuineIntel";
303     const char *brand, *fam, *mod;
304     char *last;
305     long l;

```

```

307     if ((brand = strtok_r(fsr, "-", &last)) == NULL ||
308         (fam = strtok_r(NULL, "-", &last)) == NULL ||
309         (mod = strtok_r(NULL, "-", &last)) == NULL) {
310         errx(EXIT_FAILURE, "failed to parse processor id \"%s\"", fsr);
311     }

313     if (strcmp(bstr, brand) != 0) {
314         errx(EXIT_FAILURE, "brand string \"%s\" did not match \"%s\"",
315             brand, bstr);
316     }

318     errno = 0;
319     l = strtol(fam, &last, 16);
320     if (errno != 0 || l < 0 || l >= INT_MAX || *last != '\0') {
320     if (errno != 0 || l < 0 || l >= INT_MAX || *last != '\0') {
321         errx(EXIT_FAILURE, "failed to parse family \"%s\"", fam);
322     }
323     *family = (uint_t)l;

325     l = strtol(mod, &last, 16);
326     if (errno != 0 || l < 0 || l >= INT_MAX || *last != '\0') {
326     if (errno != 0 || l < 0 || l >= INT_MAX || *last != '\0') {
327         errx(EXIT_FAILURE, "failed to parse model \"%s\"", mod);
328     }
329     *model = (uint_t)l;
330 }
  unchanged_portion_omitted

```