

new/usr/src/lib/libdemangle/common/rust.c

1

```
*****
12284 Thu Jul 11 07:13:06 2019
new/usr/src/lib/libdemangle/common/rust.c
11414 Fix smatch issue in libdemangle
*****  
_____unchanged_portion_omitted_____  
220 static boolean_t  
221 rustdem_parse_name_segment(rustdem_state_t *st, strview_t *svp, boolean_t first)  
222 {  
223     strview_t sv;  
224     strview_t name;  
225     uint64_t len;  
226     size_t rem;  
227     boolean_t last = B_FALSE;  
228  
229     if (st->rds_error != 0 || sv_remaining(svp) == 0)  
230         return (B_FALSE);  
231  
232     sv_init_sv(&sv, svp);  
233  
234     if (!rustdem_parse_num(st, &sv, &len)) {  
235         DEMDEBUG("ERROR: no leading length");  
236         st->rds_error = EINVAL;  
237         return (B_FALSE);  
238     }  
239  
240     rem = sv_remaining(&sv);  
241  
242     if (rem < len || len == SIZE_MAX) {  
243         if (rem < len || len > SIZE_MAX) {  
244             st->rds_error = EINVAL;  
245             return (B_FALSE);  
246         }  
247  
248         /* Is this the last segment before the terminating E? */  
249         if (rem == len + 1) {  
250             VERIFY3U(sv_peek(&sv, -1), ==, 'E');  
251             last = B_TRUE;  
252         }  
253  
254         if (!first && !rustdem_add_sep(st))  
255             return (B_FALSE);  
256  
257         /* Reduce length of seg to the length we parsed */  
258         (void) sv_init_sv_range(&name, &sv, len);  
259  
260         DEMDEBUG("%s: segment=%.*s", __func__, SV_PRINT(&name));  
261  
262         /*  
263          * A rust hash starts with 'h', and is the last component of a name  
264          * before the terminating 'E'  
265          */  
266         if (sv_peek(&name, 0) == 'h' && last) {  
267             if (!rustdem_parse_hash(st, &name))  
268                 return (B_FALSE);  
269             goto done;  
270         }  
271  
272         while (sv_remaining(&name) > 0) {  
273             switch (sv_peek(&name, 0)) {  
274                 case '$':  
275                     if (rustdem_parse_special(st, &name))  
276                         continue;  
277                     break;  
278                 case '_':  
279                 case ':':  
280                 case '#':  
281                 case '<':  
282                 case '>':  
283                 case '<=':  
284                 case '>=':  
285                 case '<>':  
286                 case '<*>':  
287                 case '>*>':  
288                 case '<*>=':  
289                 case '>*>=':  
290                 case '<*>*>':  
291                 case '>*>*>':  
292                 case '<*>*>*>':  
293                 case '>*>*>*>':  
294                 case '<*>*>*>*>':  
295                 case '>*>*>*>*>':  
296                 case '<*>*>*>*>*>':  
297                 case '>*>*>*>*>*>':  
298                 case '<*>*>*>*>*>*>':  
299                 case '>*>*>*>*>*>*>':  
300                 case '<*>*>*>*>*>*>*>':  
301                 case '>*>*>*>*>*>*>*>':  
302                 case '<*>*>*>*>*>*>*>*>':  
303                 case '>*>*>*>*>*>*>*>*>':  
304                 case '<*>*>*>*>*>*>*>*>*>':  
305                 case '>*>*>*>*>*>*>*>*>*>':  
306             }  
307             if (sv_peek(&name, 1) == '$') {  
308                 /* Only consume/ignore '_'. Leave  
309                  * $ for next round.  
310                  */  
311                 sv_consume_n(&name, 1);  
312                 continue;  
313             }  
314             break;  
315         }  
316         case '.':  
317             /* Convert '...' to ':::' */  
318             if (sv_peek(&name, 1) != '.')  
319                 break;  
320  
321             if (!rustdem_add_sep(st))  
322                 return (B_FALSE);  
323  
324             sv_consume_n(&name, 2);  
325             continue;  
326         default:  
327             break;  
328         }  
329  
330         if (custr_appendc(st->rds_demangled,  
331                         sv_consume_c(&name)) != 0) {  
332             st->rds_error = ENOMEM;  
333             return (B_FALSE);  
334         }  
335     }  
336  
337     done:  
338     DEMDEBUG("%s: consumed '%.*s'", __func__, (int)len, svp->sv_first);  
339     sv_consume_n(&sv, len);  
340     sv_init_sv(svp, &sv);  
341     return (B_TRUE);  
342 }  
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libdemangle/common/rust.c

2

```
278     if (sv_peek(&name, 1) == '$') {  
279         /* Only consume/ignore '_'. Leave  
280            * $ for next round.  
281            */  
282         sv_consume_n(&name, 1);  
283         continue;  
284     }  
285     break;  
286     case '.':  
287         /* Convert '...' to ':::' */  
288         if (sv_peek(&name, 1) != '.')  
289             break;  
290  
291         if (!rustdem_add_sep(st))  
292             return (B_FALSE);  
293  
294         sv_consume_n(&name, 2);  
295         continue;  
296     default:  
297         break;  
298     }  
299  
300     if (custr_appendc(st->rds_demangled,  
301                         sv_consume_c(&name)) != 0) {  
302         st->rds_error = ENOMEM;  
303         return (B_FALSE);  
304     }  
305 }  
306  
307 done:  
308 DEMDEBUG("%s: consumed '%.*s'", __func__, (int)len, svp->sv_first);  
309 sv_consume_n(&sv, len);  
310 sv_init_sv(svp, &sv);  
311 return (B_TRUE);  
312 }  
313 }  
_____unchanged_portion_omitted_____
```