Mozilla Bug 142651

1. The correlation

The Structure ReplaceData

```
1317  typedef struct ReplaceData {. .
      . . . . . .
1320JSString    *repstr;
/* replacement string */
      .. . . . . .
1327JSSubString dollarStr;
 /* for "$$" interpret_dollar result*/
} ReplaceData;
```

```
1330 static JSSubString *
1331 interpret_dollar(..., ReplaceData *rdata,...){

...
1373        rdata->dollarStr.chars = ...;
1374        rdata->dollarStr.length = 1;
1375        return &rdata->dollarStr;}
```

```
1521 do_replace(..ReplaceData *rdata,...)

1526    JSSubString *sub;
        ... ...
1536        sub = interpret_dollar(..., rdata,...);
1537        if (sub) {
1538            len = sub->length;
1539            js_strncpy(chars, sub->chars, len);
```

Conclusions

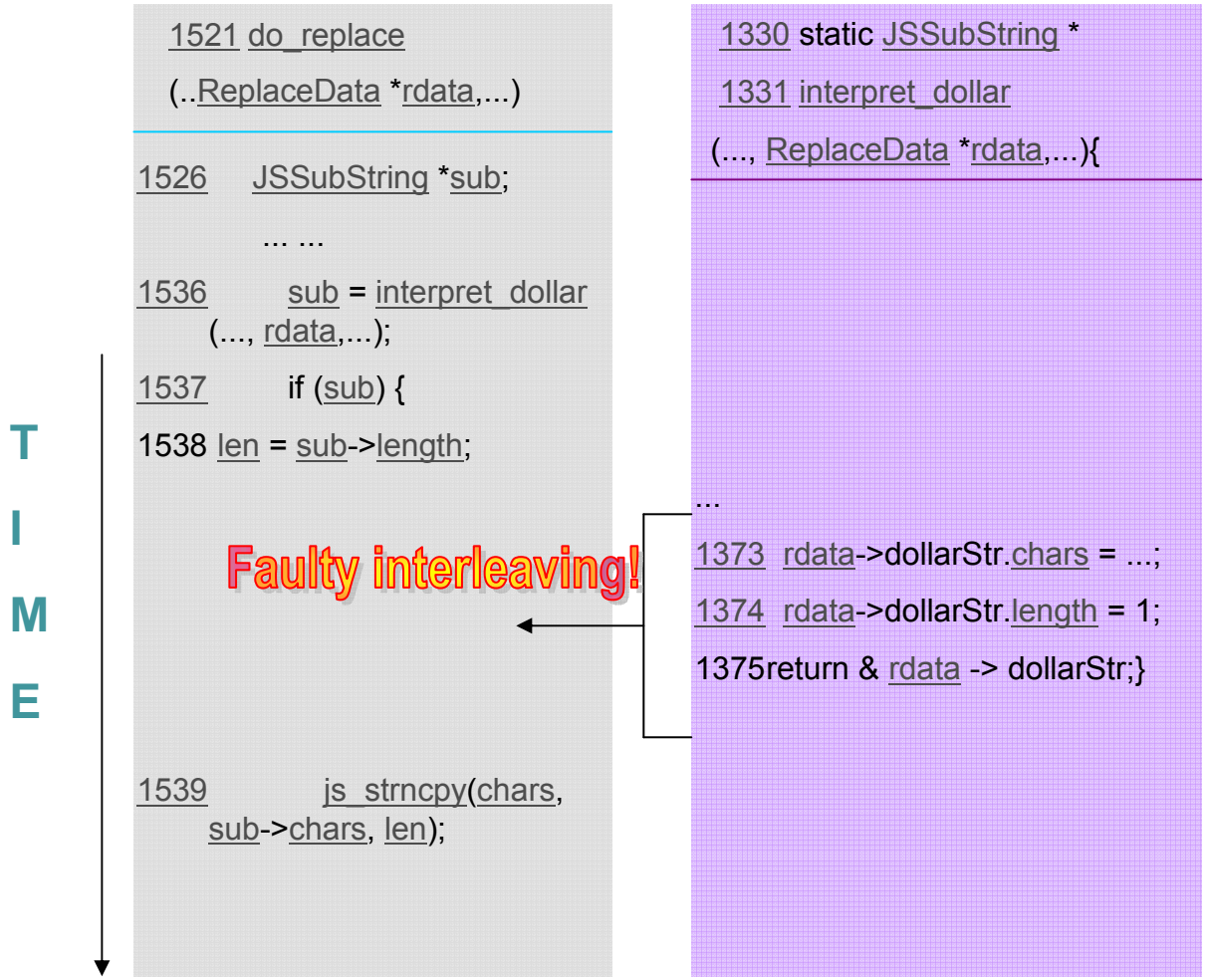
The length field for JSSubString indicates the # of characters of the chars field. Therefore, they are **correlated**.

The calculated confidence of the correlation for chars is 100% and for length is 80%.

All functions described here are in Mozilla/js/src/jsstr.c

Bug 142651 - continuation

2. The bug



What happens

This is one possible scenario, these two functions can race in other ways. Because of the interleaving *len* will not be updated and it will be copied more from *sub->chars,* which can cause crash if the later is null.

Conclusions

This is interleaving of the form thread1: read, thread2: write write, thread1: read.

The patch ensures that the two sequences are executed atomically.