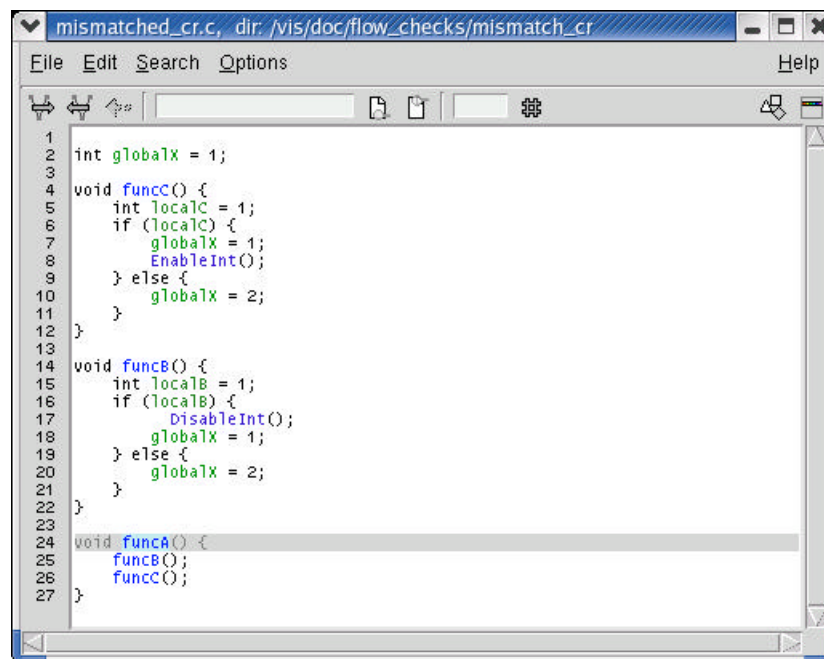


Mismatched Critical Regions

Potential issues involving the enable/disable interrupt calls or semaphores within tasks are detected by the Mismatched Critical Regions report. The analysis identifies unmatched beginnings or ends of the critical (protected) regions. The report also locates areas where there is ambiguity about whether or not a given line of code is in a critical region. Such issues leave a software system vulnerable to unprotected interrupts, and may indicate other logic problems.

Critical regions are defined by specifying a starting and ending function call. For each starting function call found in the project, the report will check through all execution paths to the end of the program or task to find a matching ending function call. If some are missing, the report will indicate the starting call location and the closest path that is missing an ending call. In a second step, the report starts from each ending function call and examines all execution paths backwards from there to find matching starting function calls until it reaches the beginning of the program or task. If not found, the report will indicate the ending function call location and the closest path missing a starting call.

The ambiguity occurs when the protected status of a line of source code depends on the path taken to that line. Note that the report will check all called or calling functions as part of this analysis.



```
mismatched_cr.c, dir: /vis/doc/flow_checks/mismatch_cr
File Edit Search Options Help
1 int globalX = 1;
2
3
4 void funcC() {
5     int localC = 1;
6     if (localC) {
7         globalX = 1;
8         EnableInt();
9     } else {
10        globalX = 2;
11    }
12 }
13
14 void funcB() {
15     int localB = 1;
16     if (localB) {
17         DisableInt();
18         globalX = 1;
19     } else {
20        globalX = 2;
21    }
22 }
23
24 void funcA() {
25     funcB();
26     funcC();
27 }
```

In the above example, both the entry into and exit from the critical region are problematic, as only certain of the conditional paths are protected. The resulting report indicates both issues, and points to the closest path that is missing an entry or exit. This provides a starting point for reviewing the mismatches.

