请注意，Flang 尚未准备好用于生产使用

李永泰 PLCT Lab
liyongtai@iscas.ac.cn
2023年12月15日

目录

- 概况和现状
- SPEC 2017 测试
- 给 Flang 做一点小小的贡献

概况和现状

什么是 LLVM Flang

LLVM Flang 是 LLVM 的 Fortran 前端，始于 f18 项目，旨在解决旧的 Flang 项目的一些问题并取代之。f18 项目后来成为 LLVM 的一部分，并被改名为 Flang。

本报告的标题来自于 Flang 的 README：

> Please note that flang is not ready yet for production usage.

- LLVM Flang https://github.com/llvm/llvm-project/tree/main/flang
- Classic Flang https://github.com/flang-compiler/flang
- f18 https://github.com/flang-compiler/f18

为什么要做一个新的 Flang

Classic Flang 的主要问题：

- 处理 pr 太慢
- 代码太老
- 不太可能成为 LLVM 的一部分

Classic Flang 的支持范围：

- Fortran 2003 完全支持
- Fortran 2008 部分支持
- Fortran 2018 没有计划

新的 Flang

使用 C++ 17 从头编写，现在是是 LLVM Project 的一部分。

最初的支持目标是 Fortran 2018，并且支持 OpenMP 4.5 OpenACC 3.0

尽可能的兼容旧的 Flang

现实

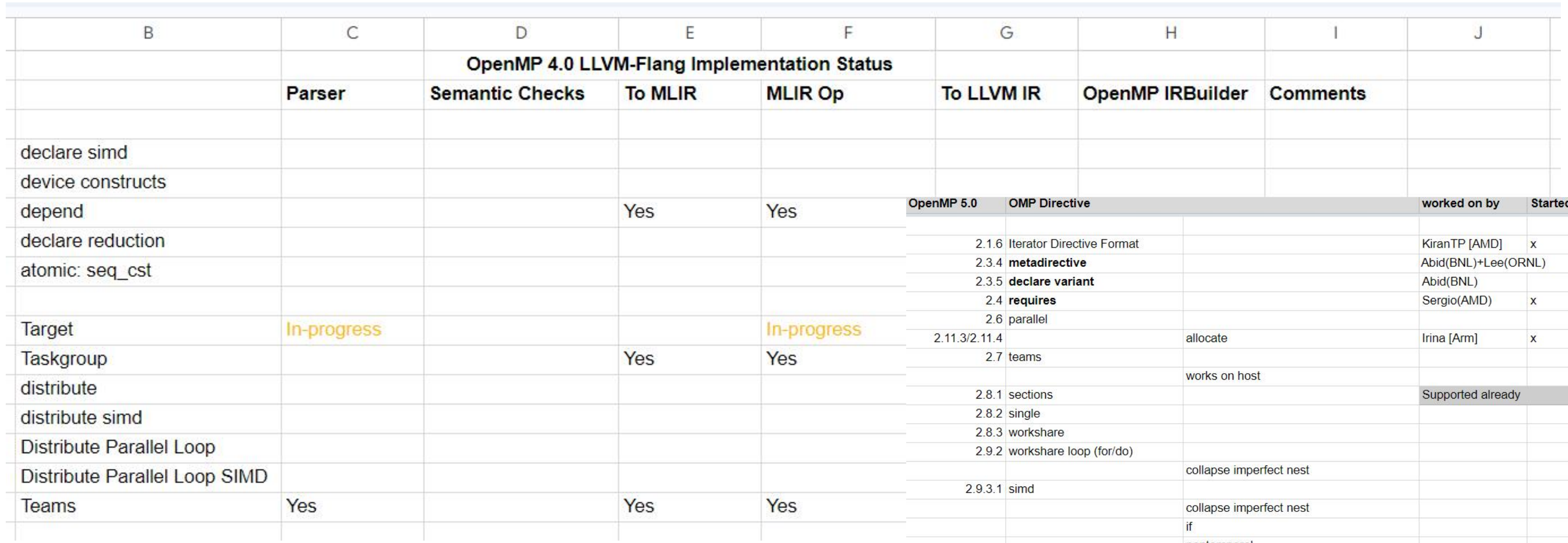
在 Github 上有 262 个未解决的相关 issue，其中近半数（117）是 OpenMP 相关。

已确认的、来自于 gfortran 测试套的 15 个问题尚未解决

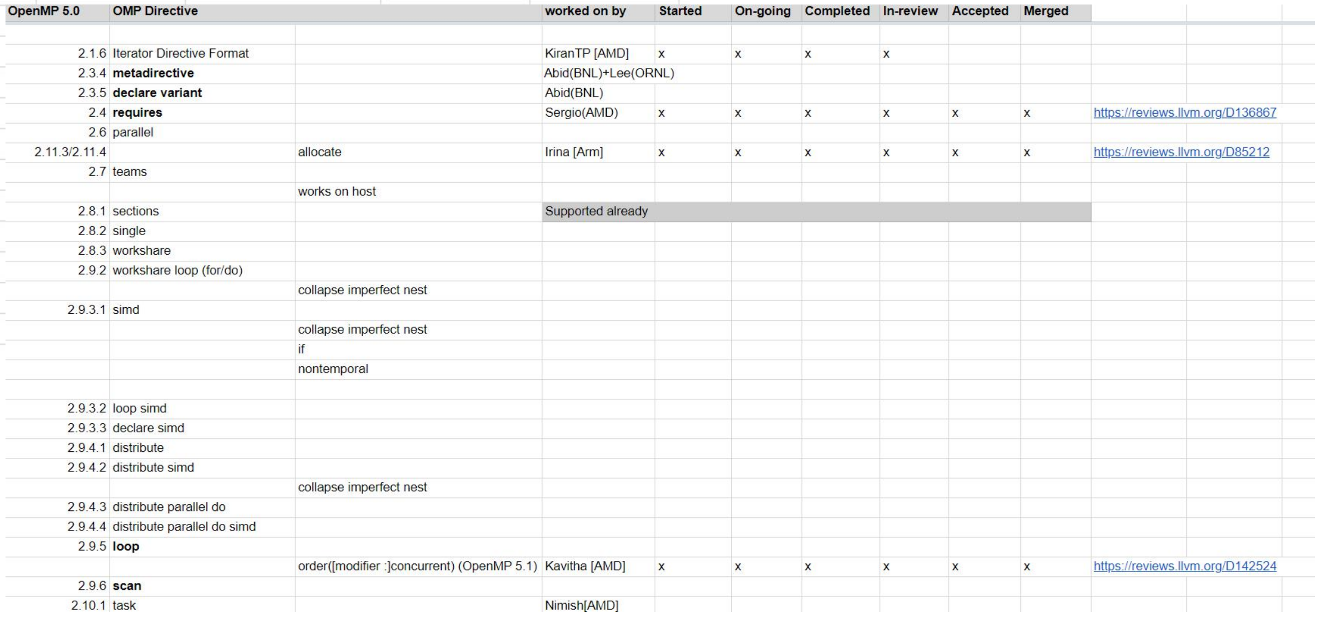
已确认的、来自于 Fujitsu 测试套的 18 个问题尚未解决

- gfortran testsuite issues https://github.com/orgs/llvm/projects/17/views/1
- Fujitsu testsuite issues https://github.com/orgs/llvm/projects/20/views/1

In-progress !



B	C	D	E	F	G	H	I	J
		OpenMP 4.0 LLVM-Flang Implementation Status						
	Parser	**Semantic Checks**	**To MLIR**	**MLIR Op**	**To LLVM IR**	**OpenMP IRBuilder**	**Comments**	
declare simd								
device constructs								
depend			Yes	Yes				
declare reduction								
atomic: seq_cst								
Target	In-progress			In-progress				
Taskgroup			Yes	Yes				
distribute								
distribute simd								
Distribute Parallel Loop								
Distribute Parallel Loop SIMD								
Teams	Yes		Yes	Yes				



OpenMP 5.0	OMP Directive		worked on by	Started	On-going	Completed	In-review	Accepted	Merged	
2.1.6	Iterator Directive Format		KiranTP [AMD]	x	x	x	x			
2.3.4	**metadirective**		Abid(BNL)+Lee(ORNL)							
2.3.5	**declare variant**		Abid(BNL)							
2.4	**requires**		Sergio(AMD)	x	x	x	x	x	x	https://reviews.llvm.org/D136867
2.6	parallel									
2.11.3/2.11.4		allocate	Irina [Arm]	x	x	x	x	x	x	https://reviews.llvm.org/D85212
2.7	teams									
		works on host								
2.8.1	sections		Supported already							
2.8.2	single									
2.8.3	workshare									
2.9.2	workshare loop (for/do)									
		collapse imperfect nest								
2.9.3.1	simd									
		collapse imperfect nest								
		if								
		nontemporal								
2.9.3.2	loop simd									
2.9.3.3	declare simd									
2.9.4.1	distribute									
2.9.4.2	distribute simd									
		collapse imperfect nest								
2.9.4.3	distribute parallel do									
2.9.4.4	distribute parallel do simd									
2.9.5	**loop**									
		order([modifier :]concurrent) (OpenMP 5.1)	Kavitha [AMD]	x	x	x	x	x	x	https://reviews.llvm.org/D142524
2.9.6	**scan**									
2.10.1	task		Nimish[AMD]							

- Flang OMP tasks https://docs.google.com/spreadsheets/d/1FvHPuSkGbl4mQZRAwCIndvQx9dQboffiD-xD0oqxgU0

SPEC 2017 测试

测试环境

licheepi 4a

revyos

flang 17

spec CPU 2017 1.0.5

测试时指定 fortran 用例使用 --size=test 参数运行最小范围的测试

具体步骤、工具集补丁详见：https://github.com/sihuan/llvm-work/tree/master/spec2017

测试结果

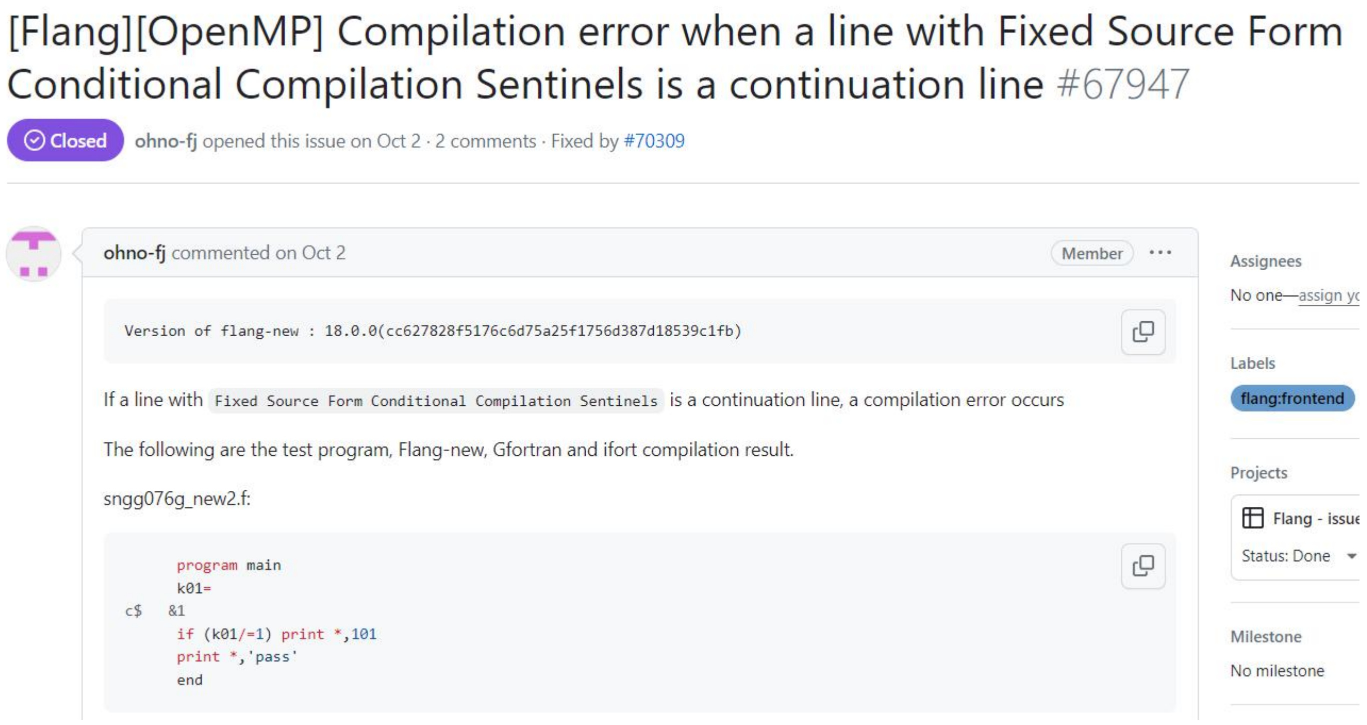
503.bwaves_r, 603.bwaves_s：编译成功但运行时段错误退出

521.wrf_r：Fortran runtime error

628.pop2_s：not yet implemented: character array expression temp with dynamic length

给 Flang 做一点小小的贡献

两次简单修复的经历



[Flang][OpenMP] Compilation error when a line with Fixed Source Form Conditional Compilation Sentinels is a continuation line #67947

Closed ohno-fj opened this issue on Oct 2 · 2 comments · Fixed by #70309

ohno-fj commented on Oct 2 Member

```
Version of flang-new : 18.0.0(cc627828f5176c6d75a25f1756d387d18539c1fb)
```

If a line with `Fixed Source Form Conditional Compilation Sentinels` is a continuation line, a compilation error occurs

The following are the test program, Flang-new, Gfortran and ifort compilation result.

sngg076g_new2.f:

```
      program main
      k01=
c$    &1
      if (k01/=1) print *,101
      print *,'pass'
      end
```

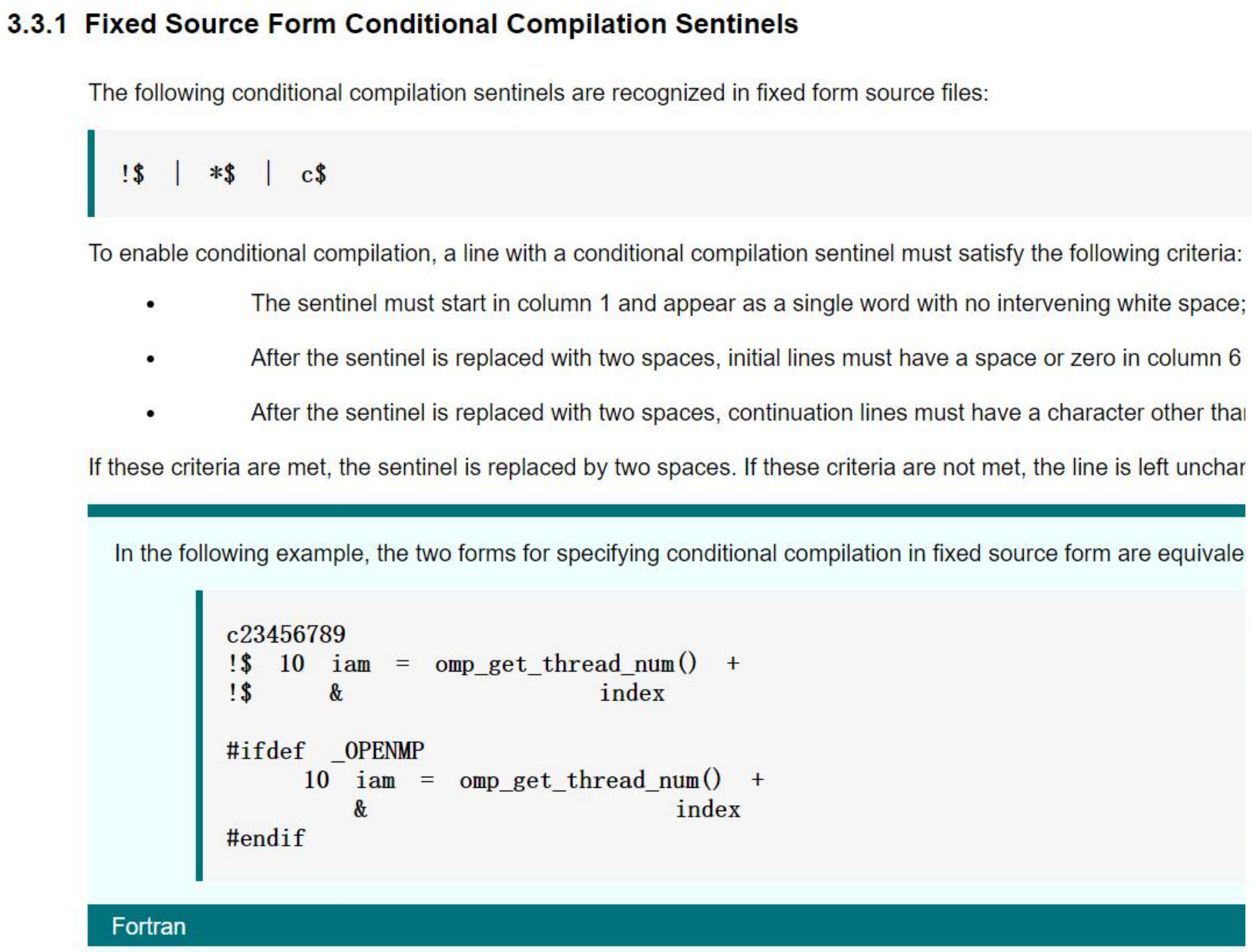
Assignees
No one—assign yo

Labels
flang:frontend

Projects
Flang - issue
Status: Done

Milestone
No milestone

https://github.com/llvm/llvm-project/issues/67947



3.3.1 Fixed Source Form Conditional Compilation Sentinels

The following conditional compilation sentinels are recognized in fixed form source files:

```
!$ | *$ | c$
```

To enable conditional compilation, a line with a conditional compilation sentinel must satisfy the following criteria:

- The sentinel must start in column 1 and appear as a single word with no intervening white space;
- After the sentinel is replaced with two spaces, initial lines must have a space or zero in column 6
- After the sentinel is replaced with two spaces, continuation lines must have a character other thai

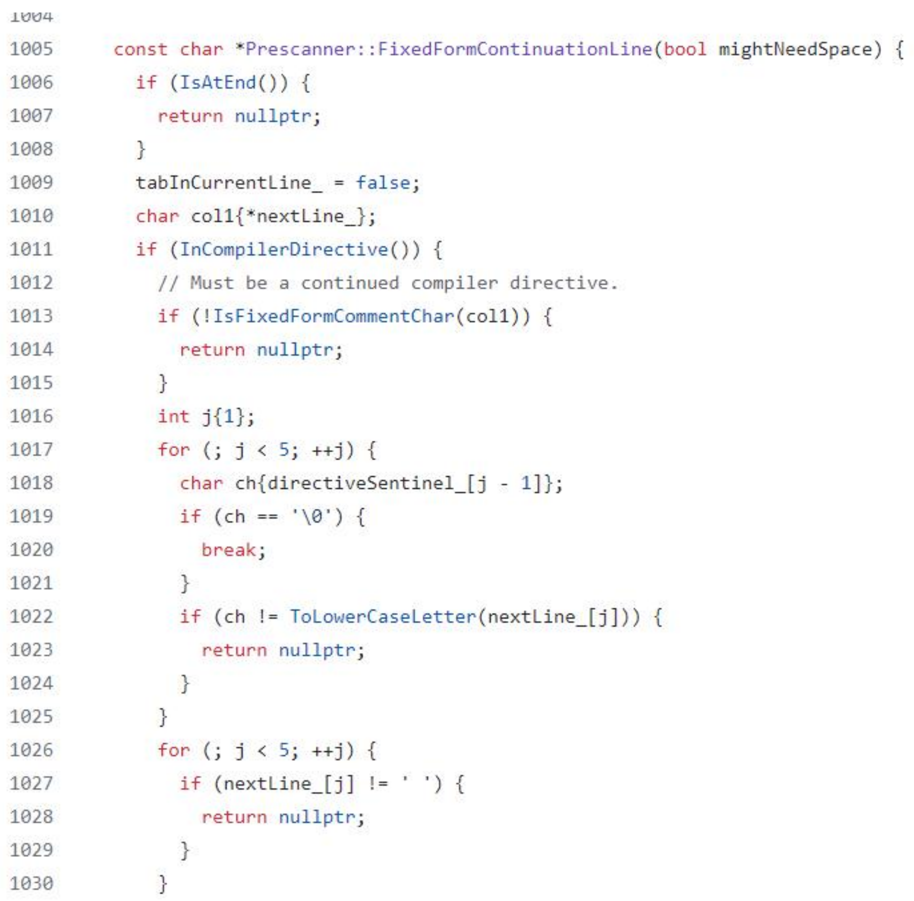
If these criteria are met, the sentinel is replaced by two spaces. If these criteria are not met, the line is left unchar

In the following example, the two forms for specifying conditional compilation in fixed source form are equivale

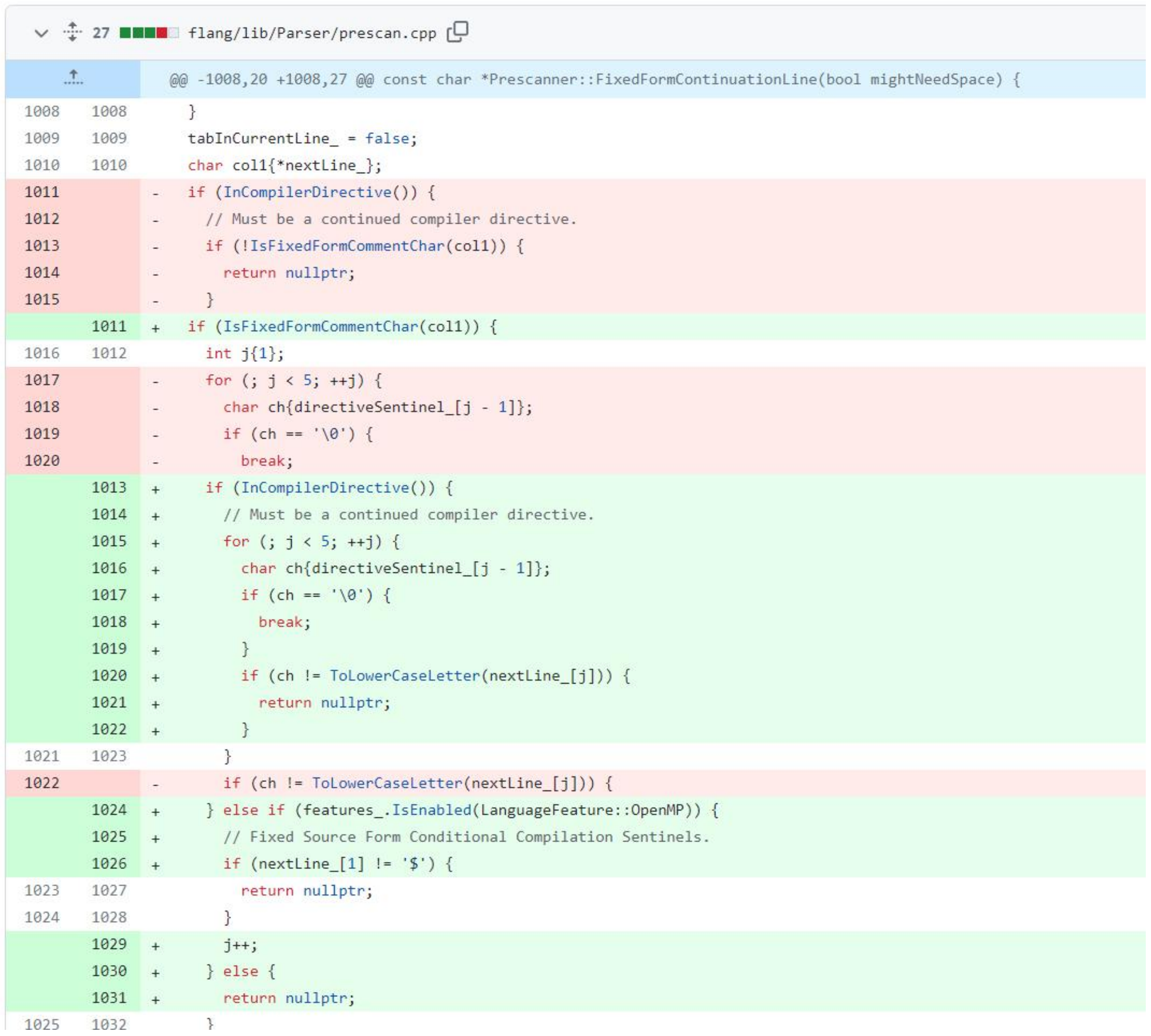
```
c23456789
!$  10  iam  =  omp_get_thread_num()  +
!$      &                   index

#ifdef  _OPENMP
      10  iam  =  omp_get_thread_num()  +
          &                    index
#endif
```

Fortran

https://www.openmp.org/spec-html/5.2/openmpsu25.html#x48-470003.3.1

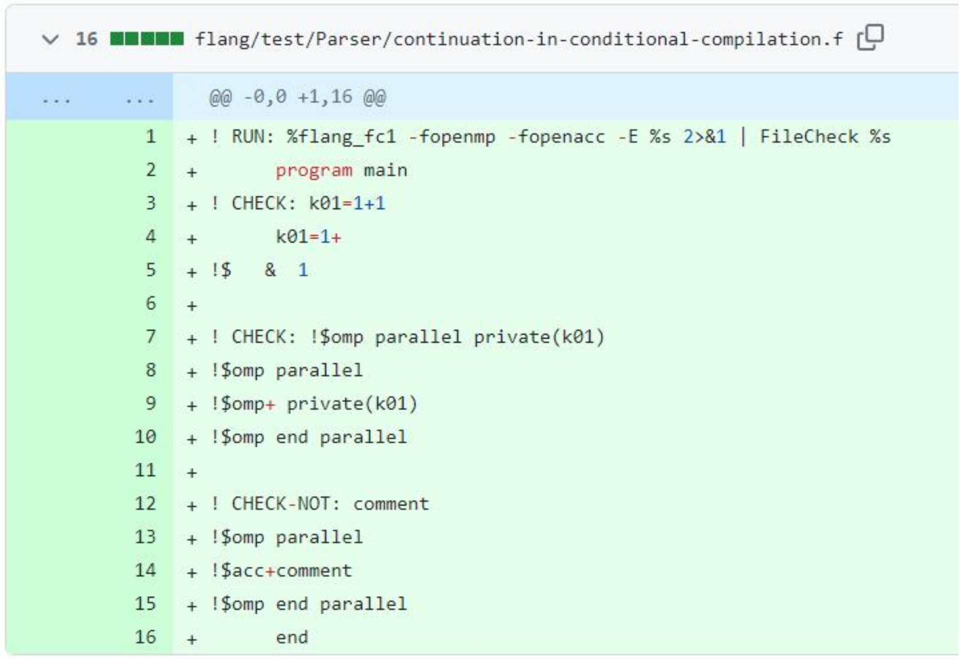


```
const char *Prescanner::FixedFormContinuationLine(bool mightNeedSpace) {
  if (IsAtEnd()) {
    return nullptr;
  }
  tabInCurrentLine_ = false;
  char col1{*nextLine_};
  if (InCompilerDirective()) {
    // Must be a continued compiler directive.
    if (!IsFixedFormCommentChar(col1)) {
      return nullptr;
    }
    int j{1};
    for (; j < 5; ++j) {
      char ch{directiveSentinel_[j - 1]};
      if (ch == '\0') {
        break;
      }
      if (ch != ToLowerCaseLetter(nextLine_[j])) {
        return nullptr;
      }
    }
    for (; j < 5; ++j) {
      if (nextLine_[j] != ' ') {
        return nullptr;
      }
    }
```



flang/lib/Parser/prescan.cpp

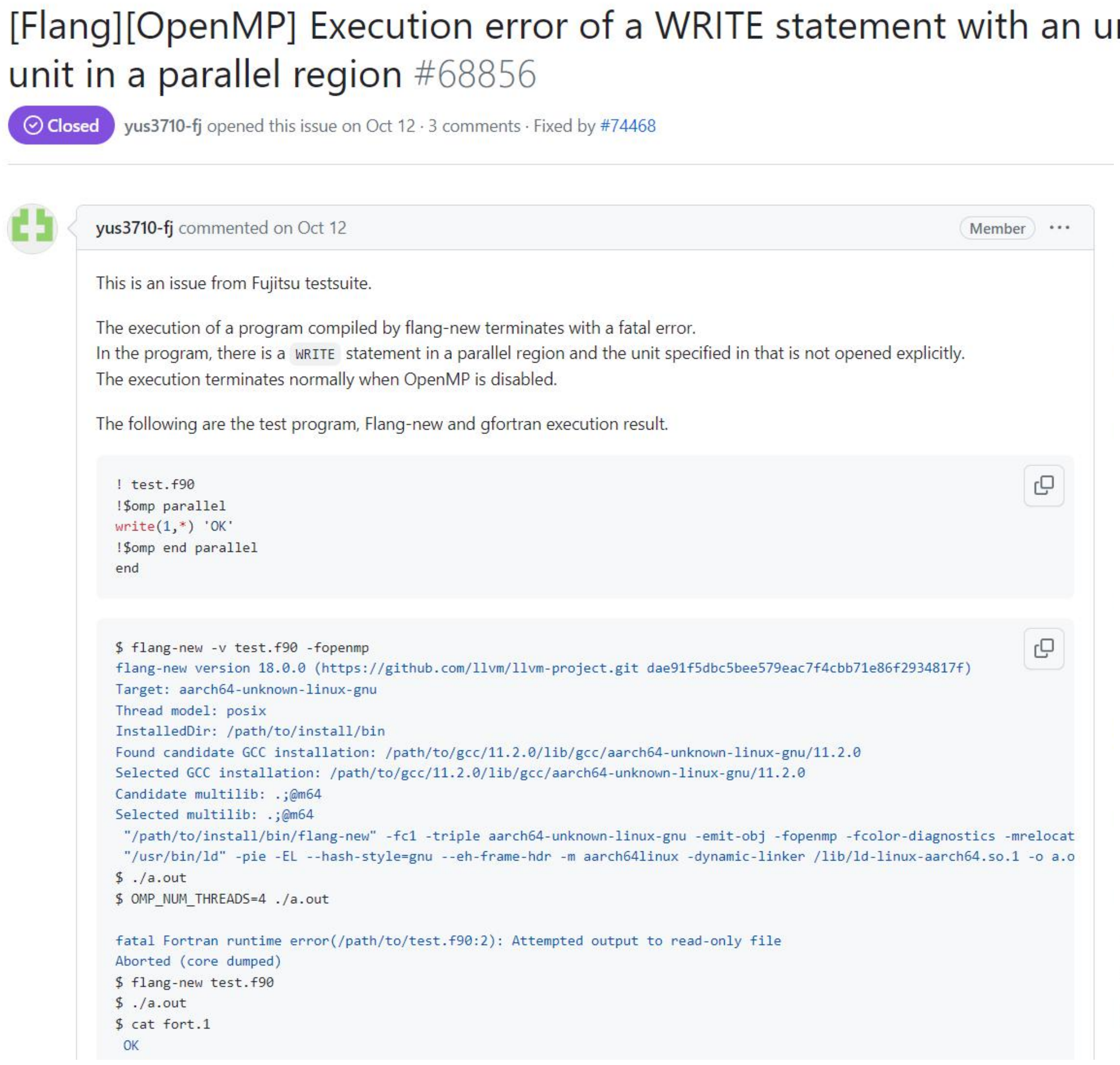
```diff
@@ -1008,20 +1008,27 @@ const char *Prescanner::FixedFormContinuationLine(bool mightNeedSpace) {
   }
   tabInCurrentLine_ = false;
   char col1{*nextLine_};
-  if (InCompilerDirective()) {
-    // Must be a continued compiler directive.
-    if (!IsFixedFormCommentChar(col1)) {
-      return nullptr;
-    }
+  if (IsFixedFormCommentChar(col1)) {
     int j{1};
-    for (; j < 5; ++j) {
-      char ch{directiveSentinel_[j - 1]};
-      if (ch == '\0') {
-        break;
+    if (InCompilerDirective()) {
+      // Must be a continued compiler directive.
+      for (; j < 5; ++j) {
+        char ch{directiveSentinel_[j - 1]};
+        if (ch == '\0') {
+          break;
+        }
+        if (ch != ToLowerCaseLetter(nextLine_[j])) {
+          return nullptr;
+        }
       }
-      if (ch != ToLowerCaseLetter(nextLine_[j])) {
+    } else if (features_.IsEnabled(LanguageFeature::OpenMP)) {
+      // Fixed Source Form Conditional Compilation Sentinels.
+      if (nextLine_[1] != '$') {
         return nullptr;
       }
+      j++;
+    } else {
+      return nullptr;
     }
```



flang/test/Parser/continuation-in-conditional-compilation.f

```diff
@@ -0,0 +1,16 @@
+! RUN: %flang_fc1 -fopenmp -fopenacc -E %s 2>&1 | FileCheck %s
+      program main
+! CHECK: k01=1+1
+      k01=1+
+!$   &  1
+
+! CHECK: !$omp parallel private(k01)
+!$omp parallel
+!$omp+ private(k01)
+!$omp end parallel
+
+! CHECK-NOT: comment
+!$omp parallel
+!$acc+comment
+!$omp end parallel
+      end
```

https://github.com/llvm/llvm-project/pull/70309



[Flang][OpenMP] Execution error of a WRITE statement with an unopened unit in a parallel region #68856

Closed yus3710-fj opened this issue on Oct 12 · 3 comments · Fixed by #74468

yus3710-fj commented on Oct 12 Member

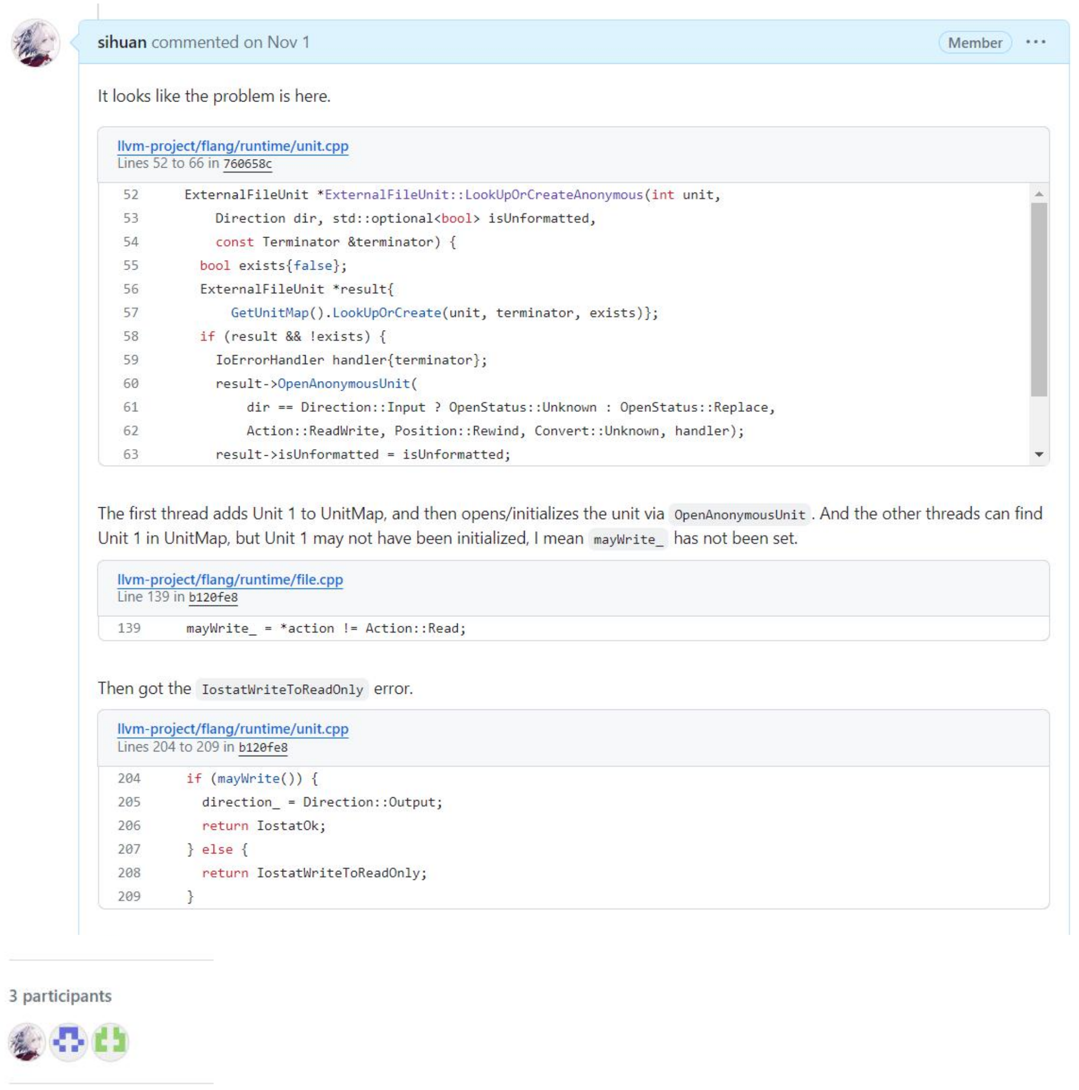
This is an issue from Fujitsu testsuite.

The execution of a program compiled by flang-new terminates with a fatal error.
In the program, there is a `WRITE` statement in a parallel region and the unit specified in that is not opened explicitly.
The execution terminates normally when OpenMP is disabled.

The following are the test program, Flang-new and gfortran execution result.

```
! test.f90
!$omp parallel
write(1,*) 'OK'
!$omp end parallel
end
```

```
$ flang-new -v test.f90 -fopenmp
flang-new version 18.0.0 (https://github.com/llvm/llvm-project.git dae91f5dbc5bee579eac7f4cbb71e86f2934817f)
Target: aarch64-unknown-linux-gnu
Thread model: posix
InstalledDir: /path/to/install/bin
Found candidate GCC installation: /path/to/gcc/11.2.0/lib/gcc/aarch64-unknown-linux-gnu/11.2.0
Selected GCC installation: /path/to/gcc/11.2.0/lib/gcc/aarch64-unknown-linux-gnu/11.2.0
Candidate multilib: .;@m64
Selected multilib: .;@m64
 "/path/to/install/bin/flang-new" -fc1 -triple aarch64-unknown-linux-gnu -emit-obj -fopenmp -fcolor-diagnostics -mrelocat
 "/usr/bin/ld" -pie -EL --hash-style=gnu --eh-frame-hdr -m aarch64linux -dynamic-linker /lib/ld-linux-aarch64.so.1 -o a.o
$ ./a.out
$ OMP_NUM_THREADS=4 ./a.out

fatal Fortran runtime error(/path/to/test.f90:2): Attempted output to read-only file
Aborted (core dumped)
$ flang-new test.f90
$ ./a.out
$ cat fort.1
 OK
```



sihuan commented on Nov 1 Member

It looks like the problem is here.

llvm-project/flang/runtime/unit.cpp
Lines 52 to 66 in 760658c

```
ExternalFileUnit *ExternalFileUnit::LookUpOrCreateAnonymous(int unit,
    Direction dir, std::optional<bool> isUnformatted,
    const Terminator &terminator) {
  bool exists{false};
  ExternalFileUnit *result{
      GetUnitMap().LookUpOrCreate(unit, terminator, exists)};
  if (result && !exists) {
    IoErrorHandler handler{terminator};
    result->OpenAnonymousUnit(
        dir == Direction::Input ? OpenStatus::Unknown : OpenStatus::Replace,
        Action::ReadWrite, Position::Rewind, Convert::Unknown, handler);
    result->isUnformatted = isUnformatted;
```

The first thread adds Unit 1 to UnitMap, and then opens/initializes the unit via `OpenAnonymousUnit`. And the other threads can find Unit 1 in UnitMap, but Unit 1 may not have been initialized, I mean `mayWrite_` has not been set.

llvm-project/flang/runtime/file.cpp
Line 139 in b120fe8

```
mayWrite_ = *action != Action::Read;
```

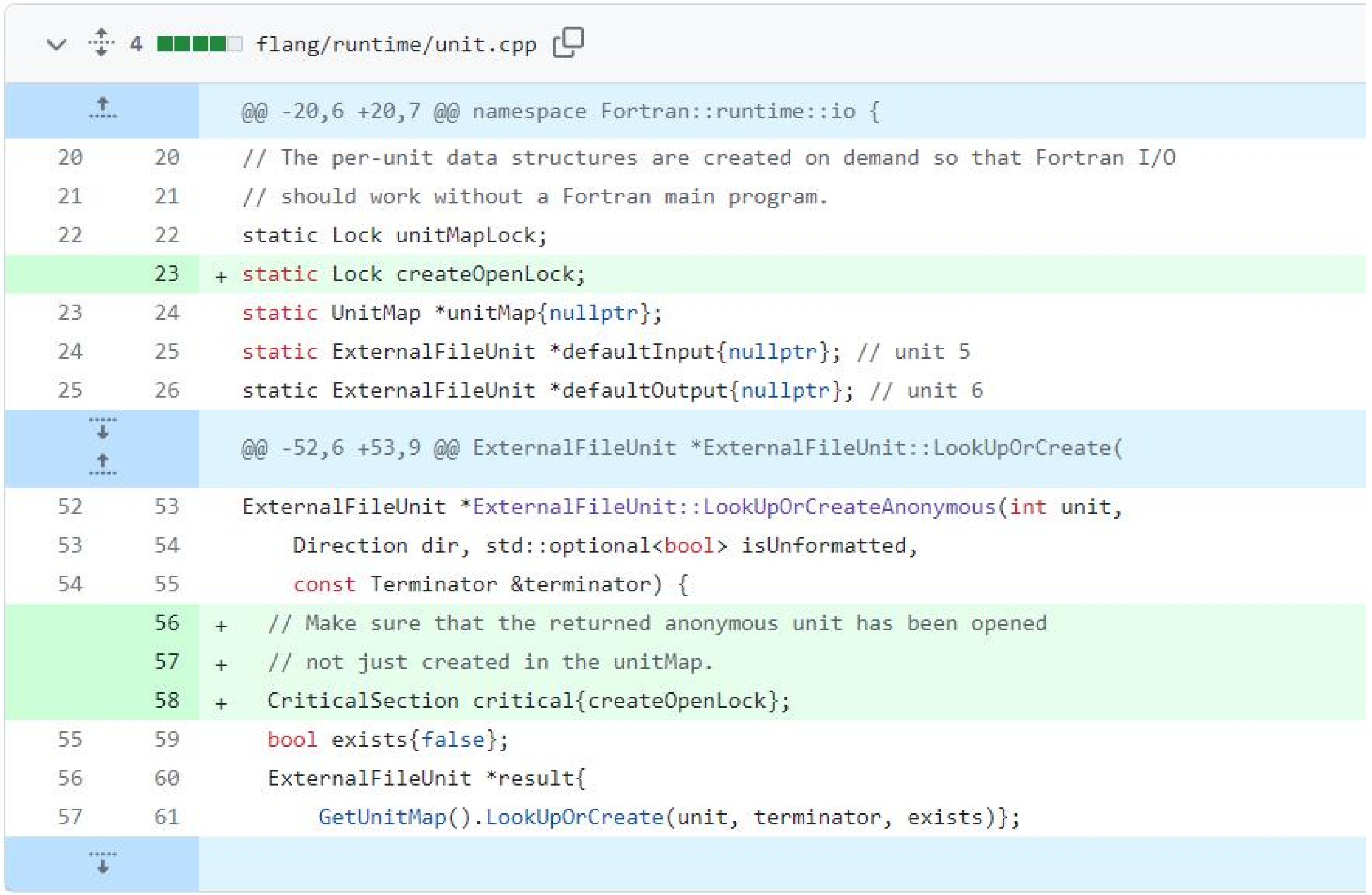
Then got the `IostatWriteToReadOnly` error.

llvm-project/flang/runtime/unit.cpp
Lines 204 to 209 in b120fe8

```
if (mayWrite()) {
  direction_ = Direction::Output;
  return IostatOk;
} else {
  return IostatWriteToReadOnly;
}
```

3 participants

https://github.com/llvm/llvm-project/issues/68856



4 flang/runtime/unit.cpp

```diff
@@ -20,6 +20,7 @@ namespace Fortran::runtime::io {
 // The per-unit data structures are created on demand so that Fortran I/O
 // should work without a Fortran main program.
 static Lock unitMapLock;
+static Lock createOpenLock;
 static UnitMap *unitMap{nullptr};
 static ExternalFileUnit *defaultInput{nullptr}; // unit 5
 static ExternalFileUnit *defaultOutput{nullptr}; // unit 6
@@ -52,6 +53,9 @@ ExternalFileUnit *ExternalFileUnit::LookUpOrCreate(
 ExternalFileUnit *ExternalFileUnit::LookUpOrCreateAnonymous(int unit,
     Direction dir, std::optional<bool> isUnformatted,
     const Terminator &terminator) {
+  // Make sure that the returned anonymous unit has been opened
+  // not just created in the unitMap.
+  CriticalSection critical{createOpenLock};
   bool exists{false};
   ExternalFileUnit *result{
       GetUnitMap().LookUpOrCreate(unit, terminator, exists)};
```

简单粗暴的上个锁
可以进一步优化

https://github.com/llvm/llvm-project/pull/74468

谢谢大家！
祝 Flang 早日删掉本报告的标题！