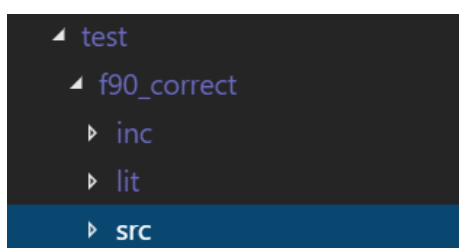
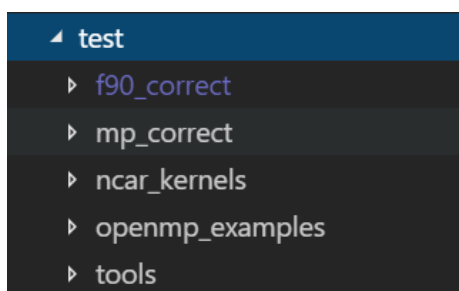


1.目录结构

test文件夹下包含对fortran分类测试的文件夹，比如f90_correct中对标准特性的测试



2.添加测试用例

f90_correct文件夹下又包含三个子文件夹：inc、lit、src，如果在f90_correct中添加一个测试用例，则需要分别在这三个文件夹下各添加一个同名的文件，例：inc/ty01.mk、lit/ty01.sh、src/ty01.f

3.编写测试用例

1) ty01.mk该文件中描述的是测试用例的编译规则和执行规则
例：

```

ty01: run

build:  $(SRC)/ty01.f
        -$(RM) ty01.$(EXESUFFIX) core *.d *.mod FOR*.DAT FTN* ftn* fort.*
        @echo ----- building test $@
        -$(CC) -c $(CFLAGS) $(SRC)/check.c -o check.$(OBJX)
        -$(FC) -c $(FFLAGS) $(LDFLAGS) $(SRC)/ty01.f -o ty01.$(OBJX)
        -$(FC) $(FFLAGS) $(LDFLAGS) ty01.$(OBJX) check.$(OBJX) $(LIBS) -o ty01.$(EXESUFFIX)

run:
        @echo ----- executing test ty01
        ty01.$(EXESUFFIX)

verify: ;

ty01.run: run

```

2) ty01.sh该文件是被测试工具lit调用的文件

例:

```

# Shared lit script for each tests. Run bash commands that run tests with make.

# RUN: KEEP_FILES=%keep FLAGS=%flags TEST_SRC=%s MAKE_FILE_DIR=%S/.. bash %S/runmake | tee %t
# RUN: cat %t | FileCheck %S/runmake

```

3) ty01.f是需要编写添加的测试用例

如果需要添加的测试用例需要执行验证测试结果，则可以调用提供的测试接口check函数，它有三个参数，计算的结果值result，期望的正确值expect及需要验证的结果数量num

例:

```
liuyao@localhost:~/work/LLVM/clang/build
type(real) :: r = 11
type(logical) :: l = .true.
type(character) :: ch = 'a'
type(double precision) :: d = 12

type(integer(1)) :: i1 = 13
type(integer(2)) :: i2 = 14
type(integer(4)) :: i4 = 15
type(integer(8)) :: i8 = 16
type(real(kind=4)) :: r4 = 17
type(real(kind=8)) :: r8 = 18
type(logical ( kind = 1 ) ) :: l1 = .true.
type(logical ( kind = 2 ) ) :: l2 = .false.
type(logical ( kind = 4 ) ) :: l4 = .true.
type(logical ( kind = 8 ) ) :: l8 = .false.
type(character (len=10,kind=1) ) :: ch10 = "ncharacter"

result(1) = i
result(2) = r
result(3) = l
result(4) = ch .eq. 'a'
result(5) = d
result(6) = i1
result(7) = i2
result(8) = i4
result(9) = i8
result(10) = r4
result(11) = r8
result(12) = l1
result(13) = l2
result(14) = l4
result(15) = l8
result(16) = ch10 .eq. "ncharacter"

data expect /10, 11, .true., .true., 12, 13, 14, 15, 16, 17, 18, .true., .false.,.true., .false., .true./
call check(result,expect,16)
```

4.回归测试

在build目录下，运行make check-all 或者make check-flang，进行回归测试。

具体测试环境的搭建可以参考王朋之前分享的文档。