

Debuggers

Dr. Hyrum Carroll

October 4, 2016

Debuggers: GDB

Two steps:

1. Compile your code using the "-g" flag
2. Execute "gdb ./programName"

Example:

1. `gfortran -g makeData.f90 -o makeData`
2. `gdb ./makeData`

GDB Commands

Command	Description
help	Display help for a command
break	Set a break point
run	Execute the program (with optional arguments)
step	Execute one line of code
list	Display source code lines
print	Display the current value of a variable
where	Display the call stack
up	Change the focus up one or more functions
down	Change the focus down one or more functions
until	Execute until a line of code (defaults to next line)
finish	Finish executing the current scope
continue	Execute until the next breakpoint
dir	Add a directory to the source file search path
<enter>	Repeat the last entered command
quit	Exit gdb

GDB

lists

```
(gdb) l
1 program makedata
2
3 implicit none
4 integer, parameter :: n = 25
5 integer :: i
6 integer, parameter :: dk = selected_real_kind(15, 307)
7 real (kind=dk) :: x, y
8 real (kind=dk) :: a1, a2, a3, a4
9 real (kind=dk) :: r
10 real (kind=dk) :: rmult
```

```
(gdb) l
11
12 call random_seed
13
14 a1 = 0.3d0
```

GDB

Break Points and Run

```
(gdb) b 20
```

```
Breakpoint 1 at 0x804887f: file makeData.f90, line 20.
```

```
(gdb) run
```

```
Starting program: /home/hcarroll/makeData
```

```
Reading symbols from shared object read from target memory
```

```
Loaded system supplied DSO at 0xb7fec000
```

```
Breakpoint 1, MAIN__ () at makeData.f90:20
```

```
20 print *, 'real multiplier'
```

```
Current language: auto; currently fortran
```

```
(gdb)
```

GDB

Stepping

```
(gdb) s
   real multiplier
21 read *, rmult
(gdb) s
3.4
23 open(unit=9, file="data2")
(gdb) s
24 do i = 1, n
(gdb)
```

GDB

continuing

```
(gdb) c
```

```
Continuing.
```

```
Program exited normally.
```

```
(gdb)
```

GDB

printing

```
(gdb) print a1
$1 = 0.299999999999999999
(gdb) print a2
$2 = -2
(gdb) print i
$3 = -1082115844
(gdb)
```


GDB

Examples

grAgnissm-withoutErrorHints.f90

errors-withoutErrorHints.f90

matrixMultiple.f90

buggy-withoutErrorHints.f90