Debugging Fx/PVM Programs

Peter A. Dinda

Compiling for debugging

Supply -g flag to fx or directly to f77:

```
fx ...other options... -f77 -g foo.fx
```

or

```
fx ...other options...
   -f77 [...other f77 options -g] foo.fx
```

or

```
f77 -g ... foo.f
```

Setting up PVM

You could install your own PVM...

Easier option, link to mine:

```
cd $HOME
mkdir pvm3 ; cd pvm3
mkdir bin; mkdir bin/ALPHA; ...
ln -s /killdeer/usr3/pdinda/pvm3/lib lib
```

Now add the following to your .cshrc:

```
setenv PVM_ROOT   $HOME/pvm3
setenv PVM_ARCH   ALPHA
setenv PVM_EXPORT DISPLAY:FX_DEBUGGER
```

Running an Fx/PVM program

Start up PVM:

```
$PVM_ROOT/lib/pvm
```

or

```
$PVM_ROOT/lib/pvmd hostfile
```

When you type “`go`”, the first copy of the program starts up the others...

PVM will try to find the other copies in

```
    $PVM_ROOT/bin/$PVM_ARCH/go
```

or an alternate path specified in the hostfile (see the PVM docs)

When you compile, you can force the executable to be placed in the right location:

```
    setenv OUTDIR $PVM_ROOT/bin/$PVM_ARCH
    fx ... -o go
```

will do the trick...

Setting up for debugging

You need to add the following to your .cshrc:

```
setenv FX_DEBUGGER gdb (or other debugger)
setenv DISPLAY mymachine:0.0
```

The environment variable `FXOPTIONS` controls the show:

For debugging:

```
setenv FXOPTIONS DEBUG
```

For tracing with XPVM:

```
setenv FXOPTIONS TRACE
```

For writing timing files when done:

```
setenv FXOPTIONS TIMING
```

Any combination:

```
setenv FXOPTIONS "DEBUG TRACE TIMING"
```

Running with debugging

xhost all the machines you are running on:

```
xhost pandora:0.0
xhost eros:0.0
```

set `FXOPTIONS` to include `DEBUG` option

Start program with

```
$FX_DEBUGGER myprog
```

Run program according to debugger... for gdb:

`run` <command line options>

Debugger windows will pop up for each of the other processes

How debugging works

When `FXOPTIONS` includes `DEBUG`, all copies of the program are spawned using PVM debugging flag

Instead of running

```
    $PVM_ROOT/bin/$PVM_ARCH/go
```

PVM starts up

```
    $PVM_ROOT/lib/debugger
        $PVM_ROOT/bin/$PVM_ARCH/go
```

The debugger script starts `$FX_DEBUGGER` on the program in an xterm set to display to `$DISPLAY`

Summary

.cshrc:

```
setenv PVM_ROOT     $HOME/pvm3
setenv PVM_ARCH     ALPHA
setenv DISPLAY      mymachine:0.0
setenv FX_DEBUGGER gdb
setenv PVM_EXPORT  DISPLAY:FX_DEBUGGER
```

Compiling:

```
fx ... -f77 -g ... -o myprogram
```

Running:

```
xhost myfirsthost
xhost mysecondhost
...

setenv FXOPTIONS DEBUG

$FX_DEBUGGER myprogram
```