Debugging the partools and parallel Packages

Norm Matloff
University of California, Davis

August 20, 2015

I place huge importance on debugging, and indeed, once wrote a book on the topic (*The Art of Debugging with GDB, DDD, and Eclipse*, N. Matloff and P. Salzman, NSP, 2008). Accordingly, **partools** includes some debugging tools, which by the way apply not only to **partools** but to the **parallel** package in general. This vignette will introduce the usage of those tools.

1 Running Example

We'll use this simple file, **fg.R** as our example:

```
f <- function(x) {
   x <- x + 1
   y <- g(x)
   x^2 + y^2
}

g <- function(t) {
   if (t > 0) return(5)
   6
}
```

2 The Fundamental Principle of Debugging

Pete Salzman and I make this statement, early in our book on debugging:

> Fixing a buggy program is a process of confirming, one by one, that the many things you *believe* to be true about the code actually *are* true. When you find that one of your assumptions is *not* true, you have found a clue to the location (if not the exact nature) of a bug.

In our code above, after executing

```
y <- g(x)
```

we may believe that **y** should be 5. But we should check that it really is 5. Eventually, we will find a case where our belief is wrong, and thus have a clue about the bug.

3 Quick and Dirty Method

The classic way to do the confirmation discussed in the last section is to add code, say that prints out the values of some key variables. This is not generally recommended, because the process of inserting such check code and later removing it is too distracting, making us lose our train of thought. It's much preferable to use a debugging aid, even a primitive one like R's **debug()**, than to insert print statements. We'll use **debug()** in the next section.

But even adding print statements, which we will do in this section, is not so easy as it sounds, because in **parallel** operations, we don't have terminal windows. A statement such as

```
cls <- makeCluster(2)
```

sets up 2 instantiations of R, but they are not attached to terminal windows, so we can't invoke **debug()**!

What we can do instead, of course, is insert code that writes confirmation messages to a file rather than to the sccreen. Even that is not so easy, because if all our cluster nodes write to the same file, we have chaos.

Instead, **partools** includes a function **dbsmsg()** that does print to a file, but with the file name having a suffix corresponding to the cluster node number. With a 2-node cluster, the file names would be **dbs.1** and **dbs.2**. We run the code, then check the files.

In the little example above, say, we add the call to **dbsmsg()** and get the code to the cluster nodes:

```
> source('fg.R')
> f
function(x) {
   x <- x + 1
   y <- g(x)
   dbsmsg(y)
   x^2 + y^2
}
> clusterExport(cls,c('f','g'))
```

Then run:

```
> clusterEvalQ(cls,f(8))
```

We then check the files **dbs.1** and **dbs.2** to see the value of **y** (both 5 in this simple example).

A similar, but somewhat more sophisticated method involves the function **dbsdump()**. We add the call to, say, **g()**, then run:

```
> clusterExport(cls,'g')
> clusterEvalQ(cls,f(8))
```

This creates one output file for each node. Let's use the one from node 2:

```
> load('last.dump.2.rda')
> debugger(last.dump.2)
Message:  Available environments had calls:
1: parallel:::.slaveRSOCK()
2: slaveLoop(makeSOCKmaster(master, port, timeout, useXDR))
3: tryCatch({
    msg <- recvData(master)
    if (msg$type == "DONE") {
        c
4: tryCatchList(expr, classes, parentenv, handlers)
5: tryCatchOne(expr, names, parentenv, handlers[[1]])
6: doTryCatch(return(expr), name, parentenv, handler)
7: tryCatch(do.call(msg$data$fun, msg$data$args, quote = TRUE), error = handle
8: tryCatchList(expr, classes, parentenv, handlers)
9: tryCatchOne(expr, names, parentenv, handlers[[1]])
10: doTryCatch(return(expr), name, parentenv, handler)
11: do.call(msg$data$fun, msg$data$args, quote = TRUE)
12: (function (expr, envir = parent.frame(), enclos = if (is.list(envir) || is.
13: eval(expr, envir, enclos)
14: f(8)
15: <tmp>#4: dbsdump()
16: eval(parse(text = cmd))
17: eval(expr, envir, enclos)

Enter an environment number, or 0 to exit  Selection: 14
Browsing in the environment with call:
   f(8)
Called from: debugger.look(ind)
Browse[1]> x
[1] 9
Browse[1]> y
[1] 5
Browse[1]> debug(g)
Browse[1]> g(x)
debugging in: g(x)
debug at <tmp>#1: {
    if (t > 0)
        return(5)
    6
}
Browse[4]> t
[1] 9
```

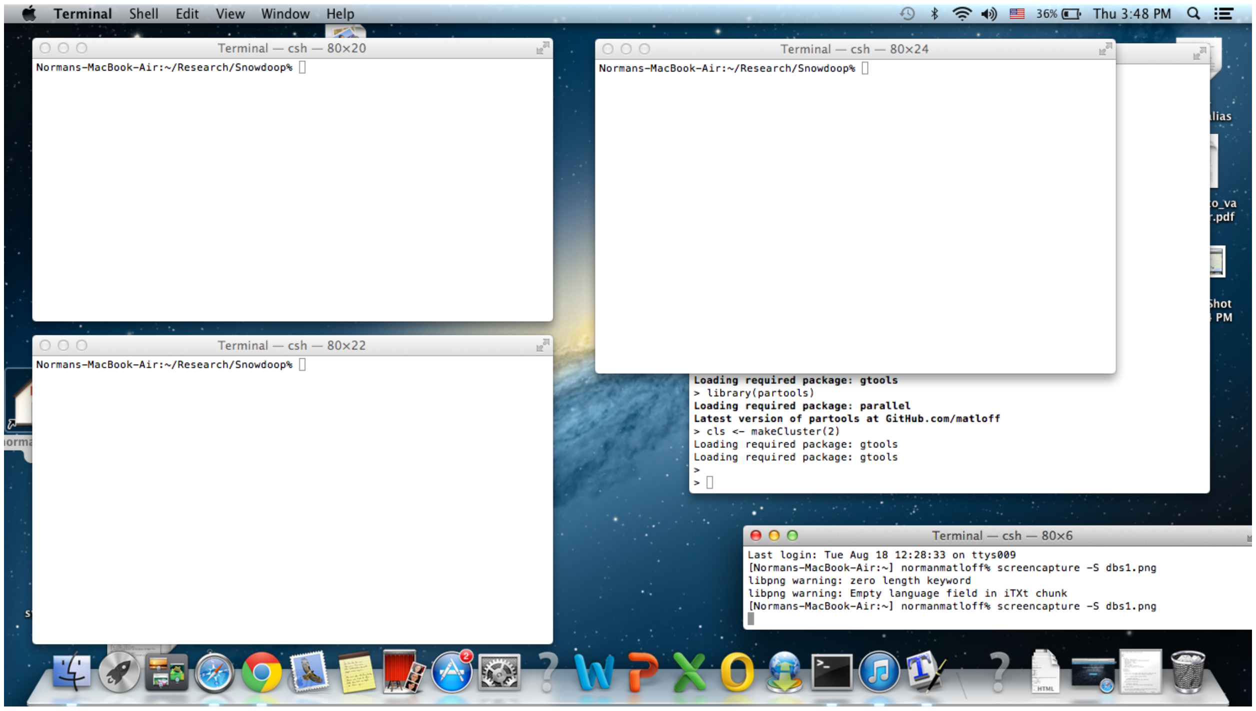
We checked the values of **x** and **y** in **f()**, and then even made a call to **g()**, checking things out there.

Again, this is an exceedingly simple example, but applying these ideas in a more realistic setting would involve the same actions.

4 Advanced Method

This method is suitable for more detailed debugging that requires single-stepping through code and so on. It is somewhat more involved, but actually very easy to use after doing it once. It requires a Unix-family system, such as a Mac or a Linux box, or Cygwin on Windows, because it make use of the **screen** utility. It allows you to debug your code with the ordinary R **debug()** facility, with one instantiation of the latter for each of your cluster nodes.

So, here is our initial screen setting:



```
Normans-MacBook-Air:~/Research/Snowdoop%
```

```
Normans-MacBook-Air:~/Research/Snowdoop%
```

```
Normans-MacBook-Air:~/Research/Snowdoop%
```

```
Loading required package: gtools
> library(partools)
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
> cls <- makeCluster(2)
Loading required package: gtools
Loading required package: gtools
>
>
```

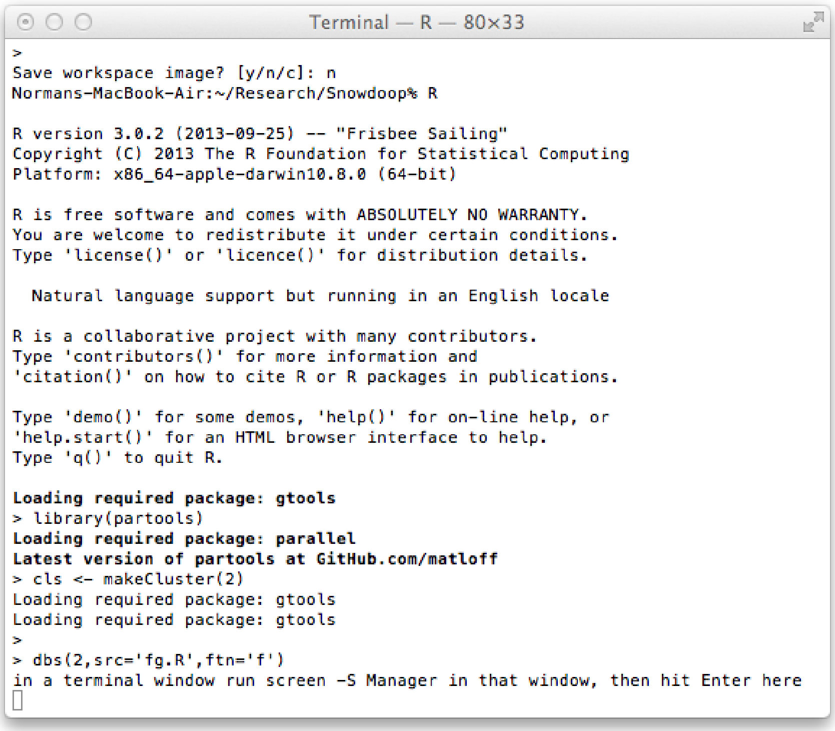
```
Last login: Tue Aug 18 12:28:33 on ttys009
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs1.png
libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs1.png
```

In preparation for running **debug()** on two cluster nodes, we have the two windows on the left. To run the "manager" node, from which **parallel** functions will be called such as

```
> clusterEvalQ(cls,f(8))
```

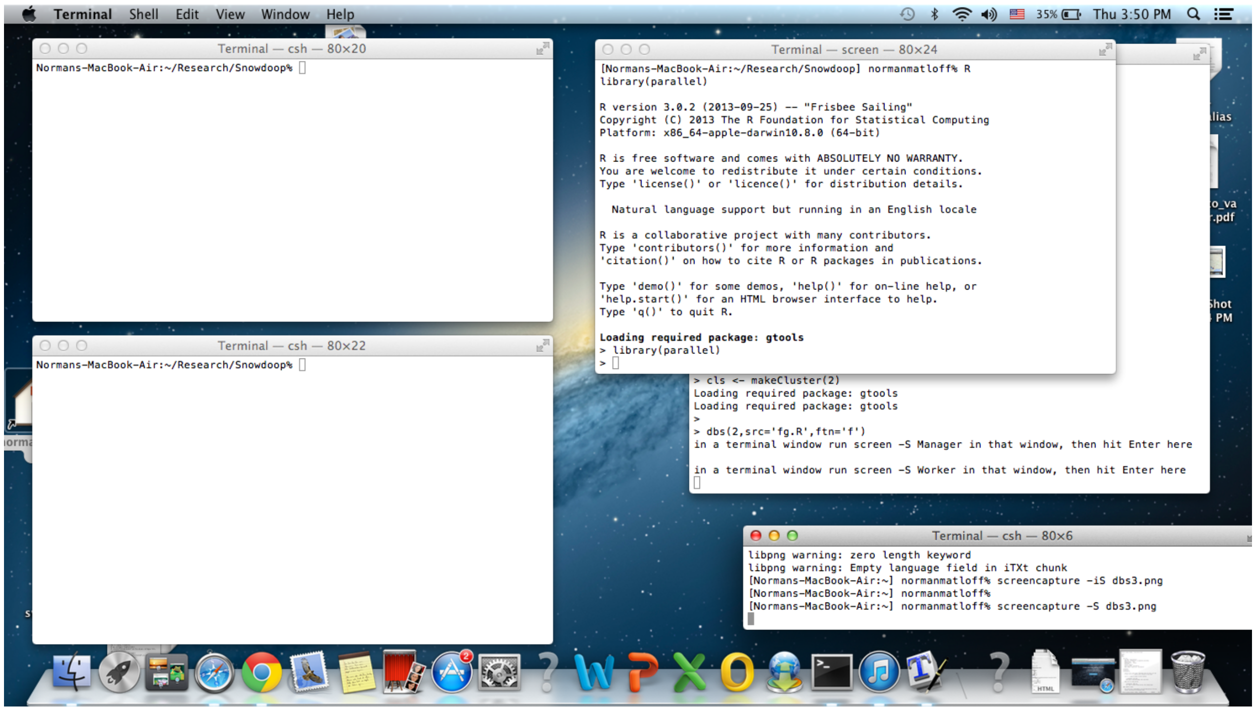
we have another window on the top right. **Under Linux, the creation of these windows can be done automatically.**

We launch the debugging from the partially hidden window:



```
>
Save workspace image? [y/n/c]: n
Normans-MacBook-Air:~/Research/Snowdoop% R

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: gtools
> library(partools)
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
> cls <- makeCluster(2)
Loading required package: gtools
Loading required package: gtools
>
> dbs(2,src='fg.R',ftn='f')
in a terminal window run screen -S Manager in that window, then hit Enter here
```

The call to **dbs()** sets up a **parallel** cluster **cls** with 2 nodes, and instructs those nodes to do **source('fg.R')** and **debug(f)**. It also instructs us, the user, to run **screen** in the manager window and in each cluster window, and it starts up R for us in the manager window:



```
Normans-MacBook-Air:~/Research/Snowdoop%
```

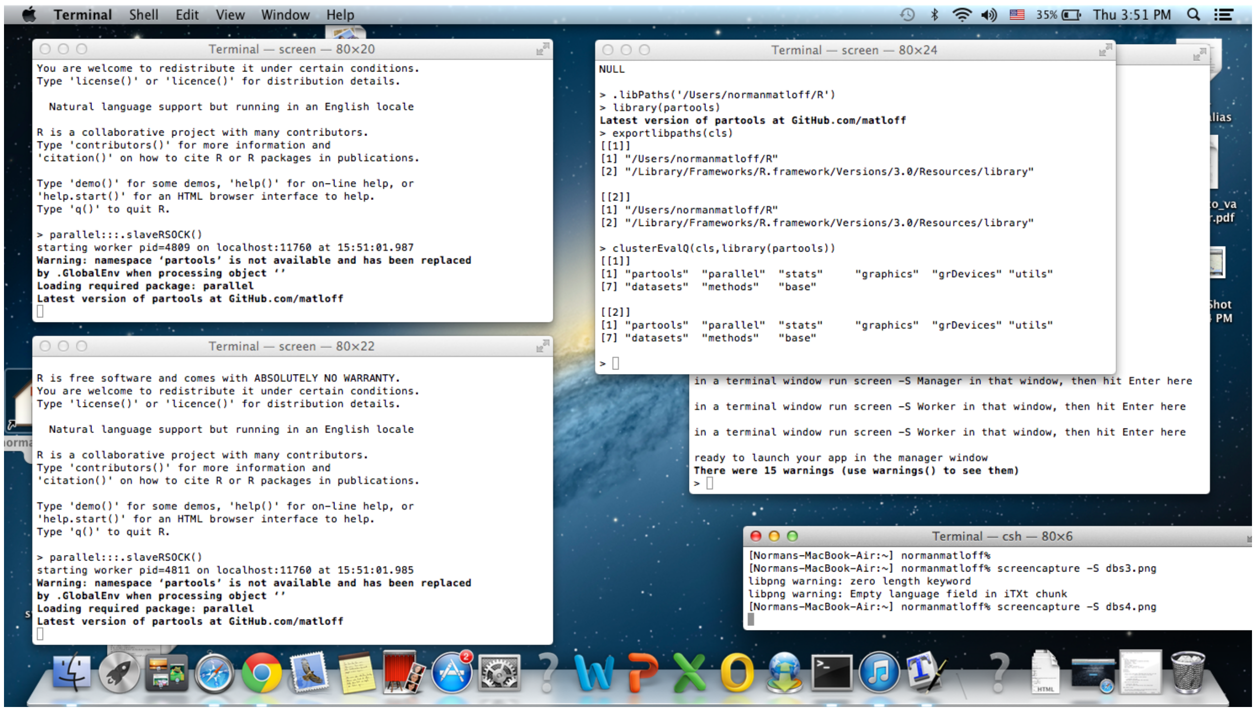
```
Normans-MacBook-Air:~/Research/Snowdoop%
```

```
[Normans-MacBook-Air:~/Research/Snowdoop] normanmatloff% R
library(parallel)

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: gtools
> library(parallel)
>
```

```
> cls <- makeCluster(2)
Loading required package: gtools
Loading required package: gtools
>
> dbs(2,src='fg.R',ftn='f')
in a terminal window run screen -S Manager in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here
```

```
libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -iS dbs3.png
[Normans-MacBook-Air:~] normanmatloff%
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs3.png
```

Finally, **dbs()** runs R in the two cluster windows. Those instantiations of R are now waiting for commands.


Terminal Shell Edit View Window Help
35% Thu 3:51 PM
Terminal — screen — 80×20
Terminal — screen — 80×22
Terminal — screen — 80×24
Terminal — csh — 80×6

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> parallel:::.slaveRSOCK()
starting worker pid=4809 on localhost:11760 at 15:51:01.987
Warning: namespace 'partools' is not available and has been replaced
by .GlobalEnv when processing object ''
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> parallel:::.slaveRSOCK()
starting worker pid=4811 on localhost:11760 at 15:51:01.985
Warning: namespace 'partools' is not available and has been replaced
by .GlobalEnv when processing object ''
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
```

```
NULL

> .libPaths('/Users/normanmatloff/R')
> library(partools)
Latest version of partools at GitHub.com/matloff
> exportlibpaths(cls)
[[1]]
[1] "/Users/normanmatloff/R"
[2] "/Library/Frameworks/R.framework/Versions/3.0/Resources/library"

[[2]]
[1] "/Users/normanmatloff/R"
[2] "/Library/Frameworks/R.framework/Versions/3.0/Resources/library"

> clusterEvalQ(cls,library(partools))
[[1]]
[1] "partools"  "parallel"  "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

[[2]]
[1] "partools"  "parallel"  "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

>
```

```
in a terminal window run screen -S Manager in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

ready to launch your app in the manager window
There were 15 warnings (use warnings() to see them)
>
```

```
[Normans-MacBook-Air:~] normanmatloff%
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs3.png
libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs4.png
```

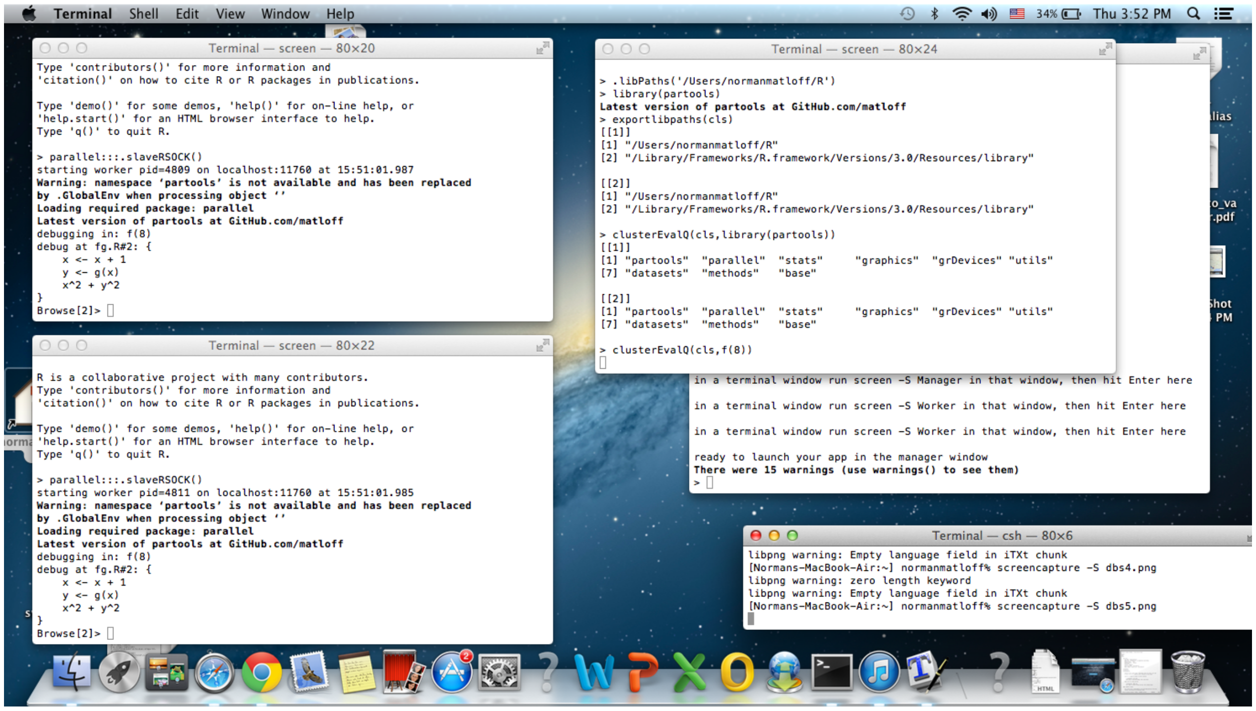
We can now run our **parallel** code normally. So, at the manager window, we type

```
> clusterEvalQ(cls,f(8))
```

And *voila!*, we are in debug mode in the two cluster node windows, with the familiar

```
Browse[]
```

prompt:


Terminal Shell Edit View Window Help
34% Thu 3:52 PM
Terminal — screen — 80×20
Terminal — screen — 80×22
Terminal — screen — 80×24
Terminal — csh — 80×6

```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> parallel:::.slaveRSOCK()
starting worker pid=4809 on localhost:11760 at 15:51:01.987
Warning: namespace 'partools' is not available and has been replaced
by .GlobalEnv when processing object ''
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
debugging in: f(8)
debug at fg.R#2: {
    x <- x + 1
    y <- g(x)
    x^2 + y^2
}
Browse[2]>
```

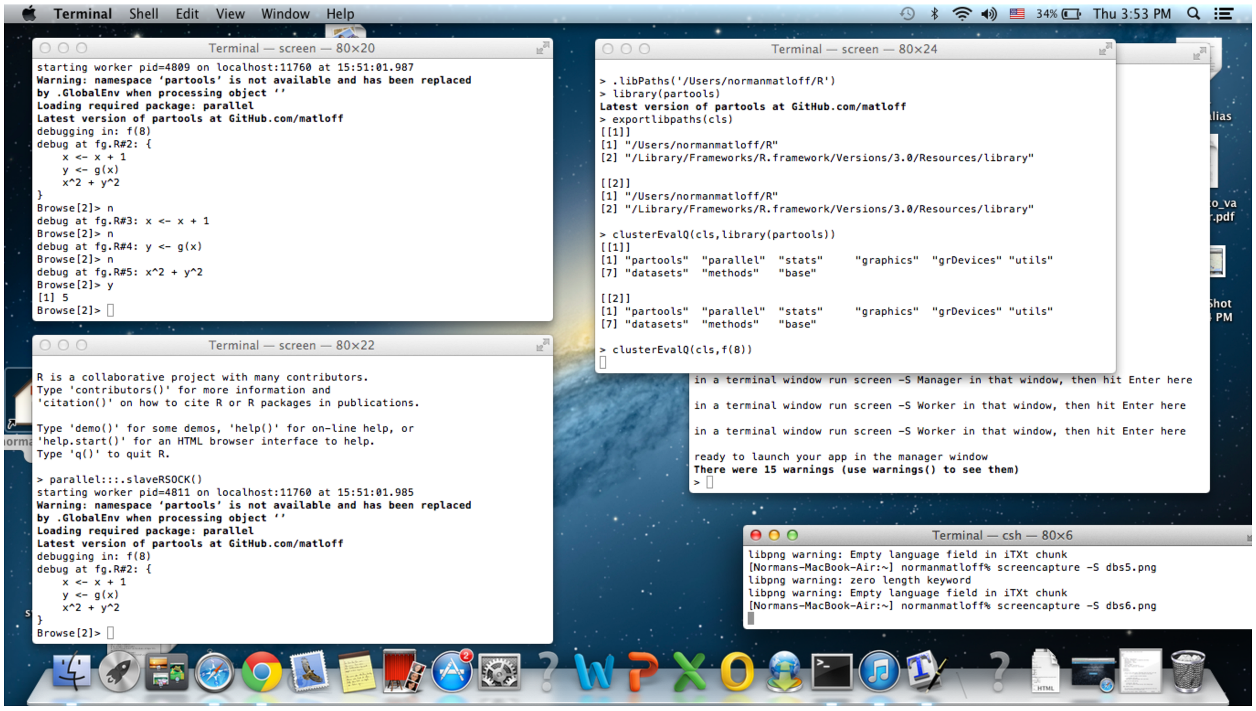
```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> parallel:::.slaveRSOCK()
starting worker pid=4811 on localhost:11760 at 15:51:01.985
Warning: namespace 'partools' is not available and has been replaced
by .GlobalEnv when processing object ''
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
debugging in: f(8)
debug at fg.R#2: {
    x <- x + 1
    y <- g(x)
    x^2 + y^2
}
Browse[2]>
```

```
> .libPaths('/Users/normanmatloff/R')
> library(partools)
Latest version of partools at GitHub.com/matloff
> exportlibpaths(cls)
[[1]]
[1] "/Users/normanmatloff/R"
[2] "/Library/Frameworks/R.framework/Versions/3.0/Resources/library"

[[2]]
[1] "/Users/normanmatloff/R"
[2] "/Library/Frameworks/R.framework/Versions/3.0/Resources/library"

> clusterEvalQ(cls,library(partools))
[[1]]
[1] "partools"  "parallel"  "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

[[2]]
[1] "partools"  "parallel"  "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

> clusterEvalQ(cls,f(8))
```

```
in a terminal window run screen -S Manager in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

ready to launch your app in the manager window
There were 15 warnings (use warnings() to see them)
>
```

```
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs4.png
libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs5.png
```

Again, we can then single-step etc. normally, say in the first cluster node window:


Terminal Shell Edit View Window Help
34% Thu 3:53 PM
Terminal — screen — 80×20
Terminal — screen — 80×24
Terminal — screen — 80×22
Terminal — csh — 80×6

```
starting worker pid=4809 on localhost:11760 at 15:51:01.987
Warning: namespace ‘partools’ is not available and has been replaced
by .GlobalEnv when processing object ‘’
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
debugging in: f(8)
debug at fg.R#2: {
    x <- x + 1
    y <- g(x)
    x^2 + y^2
}
Browse[2]> n
debug at fg.R#3: x <- x + 1
Browse[2]> n
debug at fg.R#4: y <- g(x)
Browse[2]> n
debug at fg.R#5: x^2 + y^2
Browse[2]> y
[1] 5
Browse[2]>
```

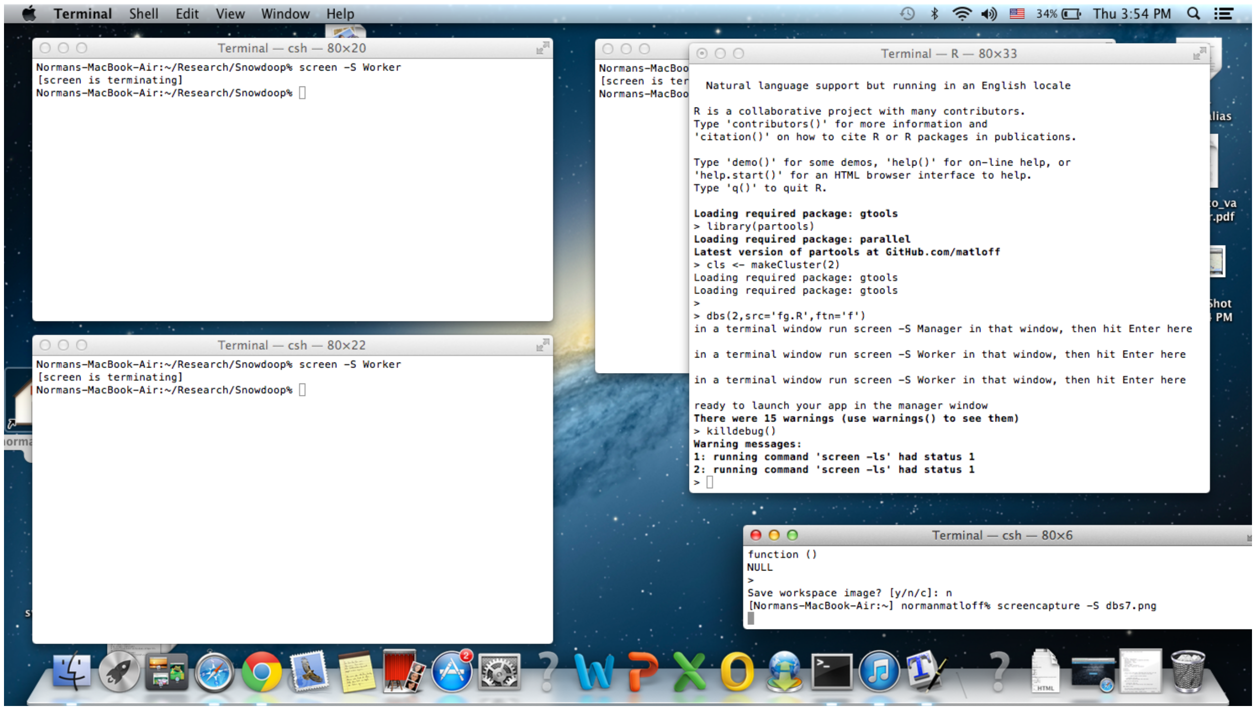
```
> .libPaths('/Users/normanmatloff/R')
> library(partools)
Latest version of partools at GitHub.com/matloff
> exportlibpaths(cls)
[[1]]
[1] "/Users/normanmatloff/R"
[2] "/Library/Frameworks/R.framework/Versions/3.0/Resources/library"

[[2]]
[1] "/Users/normanmatloff/R"
[2] "/Library/Frameworks/R.framework/Versions/3.0/Resources/library"

> clusterEvalQ(cls,library(partools))
[[1]]
[1] "partools"  "parallel"  "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

[[2]]
[1] "partools"  "parallel"  "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

> clusterEvalQ(cls,f(8))
```

```
in a terminal window run screen -S Manager in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

ready to launch your app in the manager window
There were 15 warnings (use warnings() to see them)
>
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> parallel:::.slaveRSOCK()
starting worker pid=4811 on localhost:11760 at 15:51:01.985
Warning: namespace ‘partools’ is not available and has been replaced
by .GlobalEnv when processing object ‘’
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
debugging in: f(8)
debug at fg.R#2: {
    x <- x + 1
    y <- g(x)
    x^2 + y^2
}
Browse[2]>
```

```
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs5.png
libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs6.png
```

Last, when we are done with this debugging sequence, we can kill the **screen** invocations and so on, by calling **killdebug()**:



```
Normans-MacBook-Air:~/Research/Snowdoop% screen -S Worker
[screen is terminating]
Normans-MacBook-Air:~/Research/Snowdoop%
```

```
Normans-MacBook-Air:~/Research/Snowdoop% screen -S Worker
[screen is terminating]
Normans-MacBook-Air:~/Research/Snowdoop%
```

```
Normans-MacBoo
[screen is ter
Normans-MacBoo
```

```
 Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: gtools
> library(partools)
Loading required package: parallel
Latest version of partools at GitHub.com/matloff
> cls <- makeCluster(2)
Loading required package: gtools
Loading required package: gtools
>
> dbs(2,src='fg.R',ftn='f')
in a terminal window run screen -S Manager in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

in a terminal window run screen -S Worker in that window, then hit Enter here

ready to launch your app in the manager window
There were 15 warnings (use warnings() to see them)
> killdebug()
Warning messages:
1: running command 'screen -ls' had status 1
2: running command 'screen -ls' had status 1
>
```

```
function ()
NULL
>
Save workspace image? [y/n/c]: n
[Normans-MacBook-Air:~] normanmatloff% screencapture -S dbs7.png
```