

Experimenting with a TTY Connection for R

Matthew S. Shotwell

Vanderbilt University
Department of Biostatistics

August 16, 2011

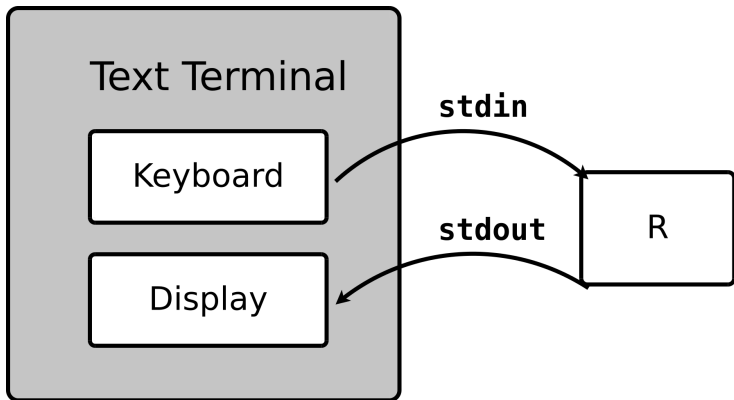


What is a TTY?

A TTY, or *computer terminal* is a two-way asynchronous communications channel with configurable properties. The name “TTY” derives from the **teletype** or text telephone.



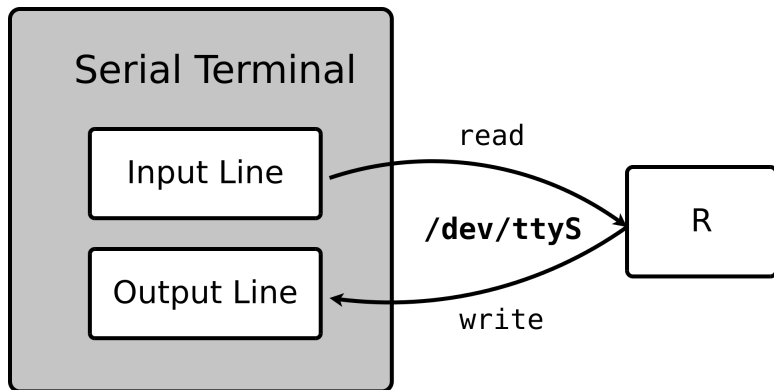
Text Terminal



configurable:

- ▶ line endings: `'\n'` vs. `'\n\r'`
- ▶ keyboard interrupts: `ctrl-c` → `SIGINT`

Serial Terminal



configurable:

- ▶ serial protocol: baud rate, character size, stop bits,
- ▶ serial protocol: parity, flow control

POSIX

- ▶ **P**ortable **O**perating **S**ystem **I**nterface for Unix
- ▶ “POSIX” suggested by Richard Stallman (*useR!* 2010 Invitee)
- ▶ [IEEE and The Open Group, 2008]
- ▶ Defines a standard and API for OS interface
 - ▶ Character Set & Locale
 - ▶ Environment Variables
 - ▶ Headers
 - ▶ General Terminal Interface

The General Terminal Interface shall be supported on any asynchronous communications ports if the implementation provides them.



OS Support for the General Terminal Interface

- ▶ Linux and UN*X (native)
- ▶ Mac OS X (native)
- ▶ Microsoft Windows (indirect)



Implementing the General Terminal Interface in R

Strategy:

- ▶ specify a new type of R *connection*
- ▶ implement a `tty` function
- ▶ configure the TTY using arguments to `tty`

Rationale:

- ▶ parsimonious with the R/S concept of IO
- ▶ utilize generic functionality (`readBin`, `flush`, *etc.*)



Implementing a New R Connection

The R connection internal code is NOT available to package developers. Hence, new R connection implementations

- ▶ cannot be in an R package
- ▶ must patch the R source code
- ▶ cannot be distributed via the CRAN

The R connection internals (at R 2.12.0) are detailed in an unofficial collection of notes: *R Connection Internals* [Shotwell, 2010] in [HTML](#) and [PDF](#). [patch + instructions](#).



The TTY Connection for R: The `tty` Function

The TTY connection patch provides:

```
tty <- function (description, open = "", blocking = TRUE, baudrate = NU
    input = NULL, output = NULL, control = NULL, local = NULL,
    chars = NULL) { ... }
```

- ▶ `input`, `output`, `control`, `local`, and `chars` are each lists of configurable TTY parameters (see `?tty` and the *General Terminal Interface* [IEEE and The Open Group, 2008])
- ▶ `tty` returns an instance of the ‘`connection`’ class



The TTY Connection for R: Text Terminal Application

```
#get a password from the terminal, don't echo characters
getpass <- function(prompt="password:") {
  cat(prompt)
  con <- tty("/dev/tty", local=list(ICANON=TRUE,ECHO=FALSE))
  pw <- readLines(con, 1)
  close(con)
  cat("\n")
  invisible(pw)
}
```

```
R> print(getpass())
password:
[1] "mysecretpassword"
```

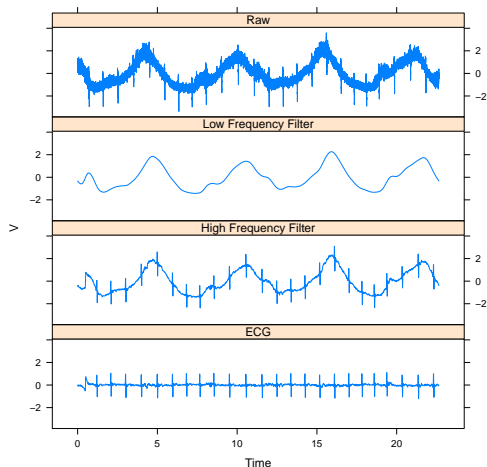


The TTY Connection for R: Serial Terminal Application

μ C Temperature Sensor Interface



The TTY Connection for R: Biomedical Application



<http://biostatmatt.com/archives/78>





IEEE and The Open Group (2008).

Standard 1003.1 & Base Specifications Issue 7.

Technical report, Institute of Electrical and Electronics Engineers & The Open Group.

URL: <http://pubs.opengroup.org/onlinepubs/9699919799/> retrieved Aug. 7, 2011.



Shotwell, M. (2010).

R connection internals.

Technical report, BioStatMatt.com.

URL: <http://biostatmatt.com/R/R-conn-ints>.



VANDERBILT
UNIVERSITY