Rxargs command design

# General & Options

Objective: A Windows command-line utility in Rust, reading a list of files (or other information) from standard input, one per line, and executes a provided command for each file.

Command to execute starts at the first arg not starting with "-", with {} replaced by one line, surrounded by quotes if arg contains spaces.

If there is no {} in command, suppose there is one as last argument (append arg to command).

If standard input is empty, do not execute command (silently unless verbose mode)

Example

rfind C:\development\Git\*\\*\*\\*.rs -exec rgrep -l pomme {} | rxargs notepad++ {}

rxargs options:  
-v verbose output, show commands that will be executed, and total counts and time

-a file, instead of reading standard input, read file. Similar to redirecting standard input, but using textautodecode crate to support most text file variants

-1, instead of executing one command per file, runs a single command at the end with all arguments (or multiple commands ensuring than arguments size does not exceed 7800 UTF-16 chars)

Possibly: -s max\_chars to control maximum arguments length for -1 option

From man xargs:

-s max-chars, --max-chars=max-chars

Use at most max-chars characters per command line, including the command and initial-arguments and the terminating nulls at the ends of the argument strings. The largest allowed value is system-dependent, and is calculated as the argument length limit for exec, less the size of your environment, less 2048 bytes of headroom. If this value is more than 128KiB, 128Kib is used as the default value; otherwise, the default value is the maximum. 1KiB is 1024 bytes. xargs automatically adapts to tighter constraints.

Possibly update rfind to call after each ActionPrint: io::stdout().flush().unwrap(); to avoid being blocked by output buffering