File Input Output

Some wrappers for standard file functions which abort the program if they fail. Useful for writing command-line utilities and small scripts, to avoid just ploughing on when a file function returns `nil`.

```
-- Find the length of a file
--   f: file name
-- returns
--   len: length of file
function lenFile(f)
  local h, len
  h = openfile(f, "r")
  len = seek(h, "end")
  closefile(h)
  return len
end

-- Guarded readfrom
--   [f]: file name
function readfrom(f)
  local h, err
  if f then h, err = %readfrom(f)
  else      h, err = %readfrom()
  end
  affirm(h, "can't read from " .. (f or "stdin") .. ": " ..
         (err or ""))
  return h
end

-- Guarded writeto
--   [f]: file name
function writeto(f)
  local h, err
  if f then h, err = %writeto(f)
  else      h, err = %writeto()
  end
  affirm(h, "can't write to " .. (f or "stdout") .. ": " ..
         (err or ""))
  return h
end
```

```
-- Guarded dofile
--   [f]: file name
function dofile(f)
  affirm(%dofile(f), "error while executing " .. f)
end

-- Guarded seek
--   f: file handle
--   w: whence to seek
--   o: offset
function seek(f, w, o)
  local ok, err
  if     o then ok, err = %seek(f, w, o)
  elseif w then ok, err = %seek(f, w)
  else          ok, err = %seek(f)
  end
  affirm(ok, "can't seek on " .. f .. ": " .. (err or ""))
end
```

more...

```
-- fileExists - return true if file exists and is readable
function fileExists(path)
  local file = openfile(path, "rb")
  if file then closefile(file) end
  return file ~= nil
end
```

FindPage · RecentChanges · preferences
edit · history
Last edited August 24, 2001 8:38 pm PDT (diff)