NAME

String::ShellQuote − quote strings for passing through the shell

SYNOPSIS

```
 $string = shell_quote @list;
 $string = shell_quote_best_effort @list;
 $string = shell_comment_quote $string;
```

DESCRIPTION

This module contains some functions which are useful for quoting strings which are going to pass through the shell or a shell-like object.

shell_quote [*string*]...

shell_quote quotes strings so they can be passed through the shell. Each *string* is quoted so that the shell will pass it along as a single argument and without further interpretation. If no *string*s are given an empty string is returned.

If any *string* can't be safely quoted **shell_quote** will **croak**.

shell_quote_best_effort [*string*]...

This is like **shell_quote**, excpet if the string can't be safely quoted it does the best it can and returns the result, instead of dying.

shell_comment_quote [*string*]

shell_comment_quote quotes the *string* so that it can safely be included in a shell-style comment (the current algorithm is that a sharp character is placed after any newlines in the string).

This routine might be changed to accept multiple *string* arguments in the future. I haven't done this yet because I'm not sure if the *string*s should be joined with blanks ($") or nothing ($,). Cast your vote today! Be sure to justify your answer.

EXAMPLES

```
 $cmd = 'fuser 2>/dev/null ' . shell_quote @files;
 @pids = split ' ', `$cmd`;

 print CFG "# Configured by: ",
           shell_comment_quote($ENV{LOGNAME}), "\n";
```

BUGS

Only Bourne shell quoting is supported. I'd like to add other shells (particularly cmd.exe), but I'm not familiar with them. It would be a big help if somebody supplied the details.

AUTHOR

Roderick Schertler <*roderick@argon.org*>

SEE ALSO

perl(1).