Which would generate a format looking something like this:

Here's a little program that's somewhat like fmt(1):

```
format =
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>
.

$/ = '';
while (<>) {
    s/\s*\n\s*/ /g;
    write;
}
```

43.2.1 Footers

While \$FORMAT_TOP_NAME contains the name of the current header format, there is no corresponding mechanism to automatically do the same thing for a footer. Not knowing how big a format is going to be until you evaluate it is one of the major problems. It's on the TODO list.

Here's one strategy: If you have a fixed-size footer, you can get footers by checking \$FORMAT_LINES_LEFT before each write() and print the footer yourself if necessary.

Here's another strategy: Open a pipe to yourself, using open(MYSELF, "|-") (see open() in *perlfunc*) and always write() to MYSELF instead of STDOUT. Have your child process massage its STDIN to rearrange headers and footers however you like. Not very convenient, but doable.

43.2.2 Accessing Formatting Internals

For low-level access to the formatting mechanism. you may use formline() and access \$^A (the \$ACCUMULATOR variable) directly.

For example:

```
$str = formline <<'END', 1,2,3;
@<<< @||| @>>>
END
```