46.2.1 The Filter

When each filter is called by Perl, a local copy of \$_ will contain the key or value to be filtered. Filtering is achieved by modifying the contents of \$_. The return code from the filter is ignored.

46.2.2 An Example – the NULL termination problem.

DBM Filters are useful for a class of problems where you *always* want to make the same transformation to all keys, all values or both.

For example, consider the following scenario. You have a DBM database that you need to share with a third-party C application. The C application assumes that *all* keys and values are NULL terminated. Unfortunately when Perl writes to DBM databases it doesn't use NULL termination, so your Perl application will have to manage NULL termination itself. When you write to the database you will have to use something like this:

```
hash{"$key\0"} = "$value\0" ;
```

Similarly the NULL needs to be taken into account when you are considering the length of existing keys/values.

It would be much better if you could ignore the NULL terminations issue in the main application code and have a mechanism that automatically added the terminating NULL to all keys and values whenever you write to the database and have them removed when you read from the database. As I'm sure you have already guessed, this is a problem that DBM Filters can fix very easily.

```
use strict;
use warnings;
use SDBM_File :
use Fcntl;
my %hash;
my $filename = "filt" ;
unlink $filename;
my $db = tie(%hash, 'SDBM_File', $filename, O_RDWR|O_CREAT, 0640)
  or die "Cannot open $filename: $!\n";
# Install DBM Filters
db \rightarrow filter_fetch_key ( sub { <math>s/0}//
db \rightarrow filter\_store\_key ( sub { $_ .= "\0" } ) ;
$db->filter_fetch_value(
    sub { no warnings 'uninitialized' ;s/\0$// } );
$db->filter_store_value( sub { $_ .= "\0" } );
$hash{"abc"} = "def" ;
my  $a = hash{"ABC"};
# ...
undef $db;
untie %hash;
```

The code above uses SDBM_File, but it will work with any of the DBM modules.

Hopefully the contents of each of the filters should be self-explanatory. Both "fetch" filters remove the terminating NULL, and both "store" filters add a terminating NULL.