

NAME

ckbox - filter for checkboxes

SYNOPSIS

```
ckbox [ -cd ] [ nr... ] < input > output
```

DESCRIPTION

ckbox is a checkbox filter. It recognizes the ASCII string '()' as an unchecked, and the string '(X)' as a checked box. This checkbox can appear anywhere on the line, but only the first occurrence on any given line will be recognized.

Without any arguments, *ckbox* will print a numbered list of all lines containing unchecked boxes from standard input. Lines with checked boxes, or lines that don't have checkboxes at all, will be ignored. The numbers represent the unique id for each checkbox, you can then run the script with a list of these numbers, separated by space, to check the checkboxes with those id numbers. When *ckbox* is used to check boxes, the entire file from standard input, with included changes, is printed.

The *-d* (done) flag reverses this behaviour somewhat, it prints checked boxes instead of unchecked boxes. However no id numbers are given, and the script does not allow you to uncheck individual checked boxes. Instead you can use the *-c* (clear) flag to uncheck all checked boxes. When used in this way, *ckbox* will print the entire file, with included changes, from standard input.

EXAMPLE

Suppose you write an email to a customer with the following contents:

Dear sir/madam

We have reviewed your request, and the deadline seems OK.
Basically the project can be divided into these steps:

- () Create a graphical design concept - outsourced
 - () Design a working front-end template
 - () Design a working back-end database

 - () Do some marketing research and testing
- PS: Or we could just release early,
and make any necessary tweaks as we go.

As for the other projects only one task remains:

- () Publish Weasel News update

Have a nice day!

With this email in place, we can do the following:

```
ckbox < email
1:  ( ) Create a graphical design concept - outsourced
2:  ( ) Design a working front-end template
3:  ( ) Design a working back-end database
4:  ( ) Do some marketing research and testing
5:  ( ) Publish Weasel News update
ckbox 1 2 5 < email | mail customer
cat email | ckbox 1 2 5 > progress
ckbox -d < progress
      (X) Create a graphical design concept - outsourced
      (X) Design a working front-end template
      (X) Publish Weasel News update
ckbox -c < progress > original
```

The progress file, and the last email sent to the customer, will look like this:

Dear sir/madam

We have reviewed your request, and the deadline seems OK.
Basically the project can be divided into these steps:

- Create a graphical design concept - outsourced
 - Design a working front-end template
 - Design a working back-end database

 - Do some marketing research and testing
- PS: Or we could just release early,
and make any necessary tweaks as we go.

As for the other projects only one task remains:

- Publish Weasel News update

Have a nice day!

Although not perfectly suited for this, *ckbox* can also be used as a 2do list for tasks that need to be repeated often:

```
ckbox < monthclean
1:   () Wash bathroom walls
2:   () Wash window frames
3:   () Vacume upholstery
4:   () Clean garden
ckbox 2 3 < monthclean > TMP && mv TMP monthclean
ckbox -c < monthclean > TMP && mv TMP monthclean
```

BUGS

If other parenthesis occur on the same line as a checkbox, the script can become confused.

The email example assumes that the customer has a user on your mainframe, if not you may need to resort to sending him a magnetic tape, or perhaps a really long serial cabel.

SEE ALSO

2do(1), *que(1)*