

Свертывание функций

Что такое "сворачивающаяся функция"?

Сворачивающаяся функция - это функция, поведение которой меняется в зависимости от обстоятельств, при которых она выполняется. Сворачивание функций полезно, когда вы обнаруживаете, что неоднократно проверяете переменную, значение которой никогда не меняется.

Как мне заставить функцию сворачиваться?

Сворачивание функций требует некоторой статической функции в среде. Типичным примером является скрипт, который дает пользователю возможность иметь "подробный" вывод.

```
#!/bin/bash

[[ $ 1 = -v || $1 = --verbose ]] && verbose=1

chatter() {
  если [[ $verbose ]]; затем
  chatter() {
    эхо "$ @"
  }
  болтовня "$ @"
  else
  chatter() {
    :
  }
fi
}

эхо "Ожидание 10 секунд".
для i в {1..10}; делать
болтовню "$ i"
сон 1
завершен
```

Как это работает?

При первом запуске `chatter()` функция переопределяет себя на основе значения `verbose` . После этого `chatter` не проверяет `$verbose`, он просто есть. Дальнейшие вызовы функции отражают ее свернутый характер. Если параметр `verbose` не установлен, `chatter` ничего не будет повторять, без дополнительных усилий со стороны разработчика.

Другие примеры

 **Fix Me!** Добавьте больше примеров!

```

# Несколько более переносимый поиск -исполняемый файл
# FIXME / НЕПРОВЕРЕННЫЙ (у меня нет доступа ко всем различным версиям
find.)
# Использование: поиск АРГУМЕНТОВ ПУТИ - используйте find как обычно,
за исключением использования -executable вместо
# различные версии -perm /+ бла-бла-бла и hacks
find() {
поиск хэша || { echo 'поиск не найден!'; выход 1; }
  # Мы можем быть уверены, что "$@" должно быть исполняемым.
  если [[ $(команда find "$@" -исполняемый файл 2> /dev/null) ]]; зат
ем
сбросить -f find #, мы можем просто использовать команду find
elif [[ $(команда find "$@" -perm /u+x 2> /dev/null) ]]; затем
find() {
введите arg args
для arg do
  [[ $arg = -исполняемый файл ]] && аргументы +=(-перм /u+x) || аргум
енты +=("$arg")
  выполненная
команда находит "${args[@]}"
}
  elif [[ $(команда find "$@" -perm +u+x 2> /dev/null) ]]; затем
find() {
набор аргументов
для аргумента do
  [[ $arg = -исполняемый файл ]] && аргументы +=(-перм + u+x) || арг
ументы +=("$arg")
  выполненная
команда находит "${args[@]}"
}
  else # Последнее средство
find() {
набор аргументов arg
для arg do
  [[ $arg = -исполняемый файл ]] && args+=(-exec test -x {} \; -print)
  || args+=("$arg")
  выполненная
команда находит "${args[@]}"
}

я нахожу "$@"
}

```



```
#!/bin/bash
# Использование сворачивающихся функций для включения / выключения со
общений отладки

[ "--debug" = "$ 1" ] && dbg=echo || dbg=:

# С этого момента, если вы используете $dbg вместо echo, вы можете вы
брать, будут ли отображаться сообщения

$dbg "Это сообщение будет отображаться только в том случае, если в ко
мандной строке указано --debug
```

Обсуждение

 [howto/collapsing_functions.txt](#)  Последнее изменение: 08.05.08 20:26 автор: bill_thomson

Этот сайт поддерживается Performing Databases - вашими экспертами по администрированию баз данных

Bash Hackers Wiki



Если не указано иное, содержимое этой вики лицензируется по следующей лицензии:
Лицензия GNU Free Documentation 1.3