

# crypt - Man Page

*функция кодирования строк (CRYPT)*

## Пролог

Эта страница руководства является частью Руководства программиста POSIX. Реализация этого интерфейса в Linux может отличаться (обратитесь к соответствующей странице руководства Linux для получения подробной информации о поведении Linux), или интерфейс может быть не реализован в Linux.

## Краткий обзор

```
#include <unistd.h>
```

```
char *crypt(const char *key, const char *salt);
```

## Описание

Функция `crypt()` – это функция кодирования строк. Алгоритм определяется реализацией.

*ключевой* аргумент указывает на строку, которая должна быть закодирована. Аргумент *salt* должен быть строкой длиной не менее двух байтов, не включая нулевой символ, выбранный из набора:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
0 1 2 3 4 5 6 7 8 9 . /
```

Первые два байта этой строки могут быть использованы для возмущения алгоритма кодирования.

Возвращаемое значение `crypt()` указывает на статические данные, которые перезаписываются каждым вызовом.

Функция `crypt()` не обязательно должна быть потокобезопасной.

## Возвращаемое значение

После успешного завершения `crypt()` возвращает указатель на закодированную строку. Первые два байта возвращаемого значения должны быть байтами аргумента. В противном случае он должен вернуть нулевой указатель и установить *errno* для указания ошибки.

## Ошибки

Функция `crypt()` завершится ошибкой, если:

# crypt - Man Page

Следующие разделы являются информативными.

## Примеры

### Кодирование паролей

Следующий пример находит запись базы данных пользователя, соответствующую определенному имени пользователя, и изменяет текущий пароль на новый пароль. Функция `crypt()` генерирует закодированную версию каждого пароля. Первый вызов `crypt()` создает закодированную версию старого пароля; затем этот закодированный пароль сравнивается с паролем, хранящимся в базе данных пользователя. Второй вызов `crypt()` кодирует новый пароль перед его сохранением.

Функция `putpwent()`, используемая в следующем примере, не является частью POSIX.1-2008.

```
#include <unistd.h>
#включить <pwd.h>
#include <string.h>
#включить <stdio.h>

...
int valid_change;
int pfd; /* Целое число для файлового дескриптора, возвращаемого open
ФАЙЛ *fpfd; /* Указатель файла для использования в putpwent(). */
struct passwd *p;
char user[100];
char oldpasswd[100];
char newpasswd[100];
char savepasswd[100];
...
valid_change = 0;
while ((p = getpwent()) != NULL) {
    * Изменить запись, если она найдена. */
    if (strcmp(p->pw_name, user) == 0) {
        if (strcmp(p->pw_passwd, crypt(oldpasswd, p->pw_passwd)) == 0) {
            strcpy(savepasswd, crypt(newpasswd, user));
            p->pw_passwd = savepasswd;
            valid_change = 1;
        }
    }
    else {
        fprintf(stderr, "Старый пароль недействителен\n");
    }
}
/* Поместить запись passwd в ptmp. */
putpwent(p, fpfd);
}
```

# crypt - Man Page

Значения, возвращаемые этой функцией, не обязательно должны быть переносимыми среди XSI-совместимых систем.

Некоторые реализации предлагают расширения через символы за пределами набора, указанного для аргумента *salt*, для указания альтернативных алгоритмов; хотя эти расширения не переносимы, они могут обеспечить лучшую безопасность. Использование *crypt()* для чего-либо, кроме хеширования паролей, не рекомендуется.

## Обоснование

Нет.

## Будущие направления

Нет.

## См. Также

`шифрование()`, `setkey()`

Базовый том определений POSIX.1-2017, `<unistd.h>`

## Авторские права

Части этого текста перепечатаны и воспроизведены в электронном виде из IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open GroupГруппа. В случае любого несоответствия между этой версией и исходным стандартом IEEE и Open Group исходный стандарт IEEE и Open Group является документом рефери. Оригинальный стандарт можно получить онлайн по адресу <http://www.opengroup.org/unix/online.html> .

Любые типографские ошибки или ошибки форматирования, которые появляются на этой странице, скорее всего, были введены во время преобразования исходных файлов в формат man page. Чтобы сообщить о таких ошибках, см. [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

## Ссылка на

`шифрование (3p)`, `setkey (3p)`, `unistd.h(0p)`.

2017 IEEE/The Open Group POSIX Programmer's Manual

# crypt - Man Page

[Главная](#) [Блог](#) [0 нас](#)