

## ВикиЧтение

### Содержание

Глава 1 Введение1.1. Модель файловой системы Linux/Unix

1.4. Почему программы GNU лучше1.4.1. Проектирование программ

1.4.3. Программирование на C

1.4.5. Заключительные соображения по поводу «GNU Coding Standards»

1.5. Пересмотренная переносимость1.6. Рекомендуемая литература1.7. Резюме

УпражненияГлава 2 Аргументы, опции и переменные окружения

2.3. Разбор опций: getopt() и getopt\_long()2.3.3.2. Длинные опции в стиле POSIX

2.4.2. Окружение в целом: environГлава 3 Управление памятью на уровне пользователя

3.2. Выделение памяти

3.2.1. Библиотечные вызовы: malloc() , calloc() , realloc() , free()

3.2.1.3. Освобождение памяти: free()

3.2.1.6. Подведение итогов из GNU Coding Standards

3.2.1.7. Использование персональных программ распределения

3.2.1.9. Только GLIBC: чтение целых строк: getline() и getdelim()

3.2.3. Системные вызовы: brk() и sbrk()

3.2.4. Вызовы ленивых программистов: alloca()3.3. Резюме

## ВикиЧтение

### Содержание

Глава 1 Введение1.1. Модель файловой системы Linux/Unix

1.4. Почему программы GNU лучше1.4.1. Проектирование программ

1.4.3. Программирование на C

1.4.5. Заключительные соображения по поводу «GNU Coding Standards»

1.5. Пересмотренная переносимость1.6. Рекомендуемая литература1.7. Резюме

УпражненияГлава 2 Аргументы, опции и переменные окружения

2.3. Разбор опций: getopt() и getopt\_long()2.3.3.2. Длинные опции в стиле POSIX

2.4.2. Окружение в целом: environГлава 3 Управление памятью на уровне пользователя

3.2. Выделение памяти

3.2.1. Библиотечные вызовы: malloc() , calloc() , realloc() , free()

3.2.1.3. Освобождение памяти: free()

3.2.1.6. Подведение итогов из GNU Coding Standards

3.2.1.7. Использование персональных программ распределения

3.2.1.9. Только GLIBC: чтение целых строк: getline() и getdelim()

3.2.3. Системные вызовы: brk() и sbrk()

3.2.4. Вызовы ленивых программистов: alloca()3.3. Резюме

## ВикиЧтение

### Содержание

Глава 1 Введение1.1. Модель файловой системы Linux/Unix

1.4. Почему программы GNU лучше1.4.1. Проектирование программ

1.4.3. Программирование на C

1.4.5. Заключительные соображения по поводу «GNU Coding Standards»

1.5. Пересмотренная переносимость1.6. Рекомендуемая литература1.7. Резюме

УпражненияГлава 2 Аргументы, опции и переменные окружения

2.3. Разбор опций: getopt() и getopt\_long()2.3.3.2. Длинные опции в стиле POSIX

2.4.2. Окружение в целом: environГлава 3 Управление памятью на уровне пользователя

3.2. Выделение памяти

3.2.1. Библиотечные вызовы: malloc() , calloc() , realloc() , free()

3.2.1.3. Освобождение памяти: free()

3.2.1.6. Подведение итогов из GNU Coding Standards

3.2.1.7. Использование персональных программ распределения

3.2.1.9. Только GLIBC: чтение целых строк: getline() и getdelim()

3.2.3. Системные вызовы: brk() и sbrk()

3.2.4. Вызовы ленивых программистов: alloca()3.3. Резюме

## ВикиЧтение

### Содержание

Глава 1 Введение1.1. Модель файловой системы Linux/Unix

1.4. Почему программы GNU лучше1.4.1. Проектирование программ

1.4.3. Программирование на C

1.4.5. Заключительные соображения по поводу «GNU Coding Standards»

1.5. Пересмотренная переносимость1.6. Рекомендуемая литература1.7. Резюме

УпражненияГлава 2 Аргументы, опции и переменные окружения

2.3. Разбор опций: getopt() и getopt\_long()2.3.3.2. Длинные опции в стиле POSIX

2.4.2. Окружение в целом: environГлава 3 Управление памятью на уровне пользователя

3.2. Выделение памяти

3.2.1. Библиотечные вызовы: malloc() , calloc() , realloc() , free()

3.2.1.3. Освобождение памяти: free()

3.2.1.6. Подведение итогов из GNU Coding Standards

3.2.1.7. Использование персональных программ распределения

3.2.1.9. Только GLIBC: чтение целых строк: getline() и getdelim()

3.2.3. Системные вызовы: brk() и sbrk()

3.2.4. Вызовы ленивых программистов: alloca()3.3. Резюме

## ВикиЧтение

### Содержание

Глава 1 Введение1.1. Модель файловой системы Linux/Unix

1.4. Почему программы GNU лучше1.4.1. Проектирование программ

1.4.3. Программирование на C

1.4.5. Заключительные соображения по поводу «GNU Coding Standards»

1.5. Пересмотренная переносимость1.6. Рекомендуемая литература1.7. Резюме

УпражненияГлава 2 Аргументы, опции и переменные окружения

2.3. Разбор опций: getopt() и getopt\_long()2.3.3.2. Длинные опции в стиле POSIX

2.4.2. Окружение в целом: environГлава 3 Управление памятью на уровне пользователя

3.2. Выделение памяти

3.2.1. Библиотечные вызовы: malloc() , calloc() , realloc() , free()

3.2.1.3. Освобождение памяти: free()

3.2.1.6. Подведение итогов из GNU Coding Standards

3.2.1.7. Использование персональных программ распределения

3.2.1.9. Только GLIBC: чтение целых строк: getline() и getdelim()

3.2.3. Системные вызовы: brk() и sbrk()

3.2.4. Вызовы ленивых программистов: alloca()3.3. Резюме