



リフト-00

あなた自身の最初のライブラリ

概要: このプロジ

ェクトは、C ライブラリのコーディングに関するものです。

これには、プログラムが依存する汎用関数が多数含まれています。

バージョン: 15

コンテンツ

私	導入	2
II	共通の指示	3
III	必須パートIII.1 III.2 Libc 関数。	4
	技術的な考慮事項。	4
	5
IV	提出と相互評価	6

第一章

導入

非常に便利な標準関数にアクセスできない場合、C プログラミングは非常に退屈になる可能性があります。このプロジェクトは、これらの機能がどのように機能するかを理解し、実装し、使用方法を学ぶことを目的としています。独自のライブラリを作成します。次のCスクールの課題で使うので助かります。

年間を通して時間をかけて libft を拡張してください。ただし、新しいプロジェクトに取り組むときは、ライブラリで使用される関数がプロジェクト ガイドラインで許可されていることを確認することを忘れないでください。

第二章

共通の指示

- プロジェクトは C で作成する必要があります。
- プロジェクトは規範に従って書かれなければなりません。ボーナス ファイル/関数がある場合、それらはノルム チェックに含まれ、内部にノルム エラーがある場合は 0 を受け取ります。
- 未定義の動作を除いて、関数は予期せず終了してはなりません (セグメンテーション違反、バス エラー、二重解放など)。これが発生した場合、プロジェクトは機能していないとみなされ、評価中に 0 が与えられます。
- ヒープに割り当てられたすべてのメモリ空間は、必要に応じて適切に解放される必要があります。漏れなし許容されるだろう。
- この作品は提出する必要がなく、採点もされませんが、プロジェクト用のテスト プログラムを作成することをお勧めします。自分の作業や同僚の作業を簡単にテストする機会が得られます。これらのテストは守備時に特に役立つことがわかります。実際、防御中に、自分のテストや評価している相手のテストを自由に使用できます。
- 割り当てられた git リポジトリに作業を送信します。git リポジトリ内の作業のみが採点されます。Deepthought があなたの作品を採点するように割り当てられている場合、それはピア評価の後に行われます。Deepthought の採点中に作業のいずれかのセクションでエラーが発生した場合、評価は停止します。

第三章

必須部分

ファイルを提出する	libft.h、ft_*.c なし
外部関数。	
説明	独自のライブラリを作成します。これは、調査に役立つツールとなる関数のコレクションです。

III.1 技術的考慮事項

- グローバル変数の宣言は禁止されています。
- より複雑な関数を分割するためにヘルパー関数が必要な場合は、それらを静的関数として定義します。
機能。こうすることで、スコープは適切なファイルに制限されます。
- すべてのファイルをリポジトリのルートに配置します。
- 未使用ファイルの持ち込みは禁止です。（以前の機能を含むファイル練習は許可されています）
- すべての .c ファイルは、フラグ -Wall -Wextra -Werror を使用してコンパイルする必要があります。

III.2 Libc 関数

libc から一連の関数をやり直す必要があります。関数は同じプロトタイプを持ち、元の関数と同じ動作を実装します。彼らは、男性の中で定義された方法に従わなければなりません。唯一の違いは名前です。これらは「ft_」プレフィックスで始まります。たとえば、strlen は ft_strlen になります。



やり直す必要がある関数のプロトタイプの中には、「restrict」修飾子を使用するものがあります。このキーワードは c99 標準の一部です。したがって、これを独自のプロトタイプに含めたり、-std=c99 フラグを使用してコードをコンパイルしたりすることは禁止されています。

以下のオリジナルの関数を実装する独自の関数を作成する必要があります。外部関数は必要ありません。

- イアルファ
- は数字です
- イサルナム
- イサスキー
- プリント
- ストレン
- メモセット
- bゼロ
- memcpy
- メモ移動
- strcpy
- strcat

第 4 章

提出と相互評価

いつものように、Git リポジトリに課題を提出します。防御中には、リポジトリ内の作業のみが評価されます。ファイル名が正しいことを確認するために、躊躇せずに再確認してください。

すべてのファイルをリポジトリのルートに配置します。