



ミニトーク

概要

このプロジェクトの目的は、UNIXシグナルを使った小さなデータ交換プログラムをコーディングすることです。

バージョン : 2

内容

I	序文	2
II	共通説明書	3
III	プロジェクト指示	5
IV	必須項目	6
V	ボーナスパート	7
VI	提出と相互評価	8

第一章 序文

シス-3-ヘキセン-1-オールは、(Z)-3-ヘキセン-1-オール、リーフアルコールとも呼ばれ、無色の油状の液体で、切ったばかりの青草や葉のような強い草緑色のにおいがします。

cis-3-Hexen-1-olは、果物や野菜のフレーバーや香水に使用される非常に重要な香気化合物である。

年間生産量は約30トン。

第二章

共通事項

- プロジェクトはC言語で書かれている必要があります。
- あなたのプロジェクトはNormに則って書かれていなければなりません。ボーナスファイル/関数がある場合、それらはノームチェックに含まれ、内部にノームエラーがある場合は0が返されます。
- 未定義の動作とは別に、関数が予期せず終了する (セグメンテーションフォールト、バスエラー、ダブルフリーなど) ことがないようにしてください。このような場合、プロジェクトは非機能とみなされ、評価時に0点が与えられます。
- ヒープで確保されたメモリ空間は、必要なときに適切に解放されなければなりません。リークは許されません。
- 対象がそれを要求している場合、ソースファイルを要求された出力にコンパイルする Makefile を `-Wall`、`-Wextra`、`-Werror` フラグで提出し、`cc` を使用し、その Makefile は再リンクしてはいけません。
- Makefileには、少なくとも`$(NAME)`、`all`、`clean`、`fclean`、および再
- プロジェクトにボーナスを回すには、Makefile にルールボーナスを含める必要があります、それによってプロジェクトのメイン部分で禁止されている様々なヘッダー、`libraries` または関数がすべて追加されます。ボーナスは、主体が何も指定しない場合は、別のファイル `_bonus.{c/h}` にする必要があります。必須部分とボーナス部分の評価は別々に行われます。
- プロジェクトで `libft` を使用できるようにする場合、そのソースと関連する Makefile を `libft` フォルダにコピーしておく必要があります。あなたのプロジェクトの Makefile は、その Makefile を使ってライブラリをコンパイルし、その後プロジェクトをコンパイルしなければなりません。
- テストプログラムは提出する必要がなく、採点もされませんが、プロジェクトのテストプログラムを作成することをお勧めします。テストプログラムを作成することで、自分の作品や仲間の作品を簡単にテストすることができます。このようなテストは、特にデフェンスの時に役立つと思います。実際、審査では、自分のテストや審査する仲間のテストを自由に使用することができます。
- 指定された `git` リポジトリに作品を提出する。`git` リポジトリにある作品だけが採点されます。`Deepthought` があなたの作品の採点を担当する場合、採点は以下のように行われます。

相互評価後Deepthoughtの採点中に、あなたの作品のいずれかのセクションでエラーが発生した場合、評価は停止されます。

第三章

プロジェクト指示

- 実行ファイルの名前をclientとserverにします。
- ソースファイルをコンパイルするMakefileを提出する必要があります。再リンクはしてはいけません。
- あなたのlibftをぜひ使ってみてください。
- エラーは徹底的に処理しなければならない。セグメンテーションエラー、バスエラー、ダブルフリーなど、プログラムが予期せず終了するようなことがあってはなりません。
- プログラムにメモリリークがあってはならない。
- グローバル変数は1つのプログラムにつき1つ（クライアント用とサーバー用）持つことができますが、その使い方を正当化する必要があります。
- 必須項目をクリアするために、以下の機能を使用することが許可されています。
 - 書く
 - ft_printf と同等の YOU コード。
 - 信号
 - シングエンプティセット
 - シガデッドセット
 - シグナメント
 - 討ち果たす
 - ゲットピッド
 - マロック
 - じゅうざんまい
 - あいだをおく
 - 寝
 - 寝
 - 出口

第四章 必須項目

クライアントとサーバーという形で通信プログラムを作成する必要があります。

- まず、サーバーを起動する必要があります。起動後、PIDを表示する必要があります。
- クライアントは2つのパラメータを受け取る。
 - サーバーのPIDです。
 - 送信する文字列。
- クライアントは、パラメータとして渡された文字列をサーバーに送信しなければならない。文字列を受信したら、サーバーはそれを印刷しなければならない。
- サーバーは、かなり速く文字列を表示しなければならない。素早くというのは、もしあなたが時間がかかりすぎると思うなら、それは長すぎるということです。



100文字表示で1秒はやりすぎ！？

- サーバーは、再起動する必要なく、複数のクライアントから連続して文字列を受信できるはずです。
- クライアントとサーバーの間の通信は、UNIXのシグナルだけで行う必要があります。
- この2つの信号しか使えません。SIGUSR1 と SIGUSR2 です。



Linuxシステムは、すでに保留中のシグナルがある場合、シグナルをキューに入れることはありません。 ボーナスタイム？

第五章 ボーナス パート

ボーナスリスト

- サーバーは受信したすべてのメッセージに対して、クライアントに信号を送り返すことで承認する。
- ユニコード文字に対応!



ボーナスパーツは、必須パーツがPERFECTである場合にのみ査定されます。パーフェクトとは、必須パートが統合的に行われ、誤動作することなく動作することを意味します。必須条件をすべてクリアしていない場合、ボーナスパーツの評価は一切行われません。

第六章

提出と相互評価

通常通り、Gitリポジトリに課題を提出する。防衛戦では、あなたのリポジトリ内の作品だけが評価されます。ファイル名が正しいかどうか、遠慮なく再確認してください。