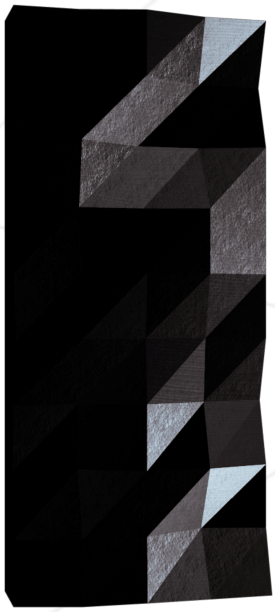

Minitalk

Resumo:

O objetivo deste projeto é codificar um pequeno programa de troca de dados usando sinais UNIX.

Versão 2

Conteúdo

Capítulo I

Prefácio

O cis-3-hexen-1-ol, também conhecido como (Z)-3-hexen-1-ol e álcool de folha, é um líquido oleoso incolor com um intenso odor verde-gramado de grama e folhas recém-cortadas.

É produzido em pequenas quantidades pela maioria das plantas e atua como um atrativo para muitos insetos predadores. cis-3-hexen-1-ol é um composto de aroma muito importante que é usado em sabores de frutas e vegetais e em perfumes.

A produção anual é de cerca de 30 toneladas.

Capítulo II

Instruções Comuns

- Seu projeto deve ser escrito em C.
- Seu projeto deve ser redigido de acordo com a Norma. Se você tiver arquivos/funções de bônus, eles estão incluídos na verificação da norma e você receberá um0se houver um erro de norma dentro.
- Suas funções não devem ser encerradas inesperadamente (falha de segmentação, erro de barramento, double free, etc.) além de comportamentos indefinidos. Caso isso aconteça, seu projeto será considerado não funcional e receberá uma0durante a avaliação.
- Todo o espaço de memória alocado no heap deve ser liberado adequadamente quando necessário. Nenhum vazamento será tolerado.
- Se o assunto exigir, você deve enviar umMakefileque compilará seus arquivos de origem para a saída necessária com os sinalizadores -Parede, -Wextrae -Erro,use cc, e seu Makefile não deve revincular.
- SeuMakefiledeve conter pelo menos as regras $(NOME), tudo, limpo, flimpoe ré.
- Para entregar bônus ao seu projeto, você deve incluir uma regrabônusao seu Makefile, que adicionará todos os vários cabeçalhos, bibliotecas ou funções proibidas na parte principal do projeto. Os bônus devem estar em um arquivo diferente _bônus.{c/h}se o sujeito não especificar mais nada. A avaliação da parte obrigatória e bônus é feita separadamente.
- Se o seu projeto permite que você use seuliberdade,você deve copiar suas fontes e seus associadosMakefileem umlibftpasta com seu Makefile associado. do seu projeto Makefile deve compilar a biblioteca usando seuMakefile,em seguida, compile o projeto.
- Nós encorajamos você a criar programas de teste para o seu projeto, mesmo que este trabalho **não terá que ser enviado e não será avaliado**. Isso lhe dará a chance de testar facilmente seu trabalho e o de seus colegas. Você achará esses testes especialmente úteis durante sua defesa. De fato, durante a defesa, você é livre para usar seus testes e/ou os testes do par que está avaliando.
- Envie seu trabalho para o repositório git atribuído. Apenas o trabalho no repositório git será avaliado. Se o Deepthought for designado para avaliar seu trabalho, isso será feito

depois de suas avaliações de pares. Se ocorrer um erro em qualquer seção do seu trabalho durante a avaliação do Deepthought, a avaliação será interrompida.

Capítulo III

instruções do projeto

- Nomeie seus arquivos executáveisclienteeservidor.
- Você tem que virar em umMakefileque irá compilar seus arquivos de origem. Não deve religar.
- Você pode definitivamente usar o seulibft.
- Você tem que lidar com os erros completamente. De forma alguma seu programa deve fechar inesperadamente (falha de segmentação, erro de barramento, double free, e assim por diante).
- Seu programa não deve ter**perda de memória**.
- Você pode ter**uma variável global por programa**(um para o cliente e outro para o servidor), mas terá de justificar a sua utilização.
- Para completar a parte obrigatória, você pode usar as seguintes funções:
  - escrever
  - ft_printfe qualquer equivalente VOCÊ codificado
  - sinal
  - sigemptyset
  - sigaddset
  - ação de adesão
  - matar
  - enlouquecer
  - malloc
  - livre
  - pausa
  - dormir
  - você dorme
  - saída

Capítulo IV

Parte Obrigatória

Você deve criar um programa de comunicação na forma de um**cliente**e um**servidor**.

- O servidor deve ser iniciado primeiro. Após o seu lançamento, tem de imprimir o seuPID.
- O cliente recebe dois parâmetros:
  - O servidorPID.
  - A string a ser enviada.
- O cliente deve enviar a string passada como parâmetro para o servidor. Uma vez recebida a string, o servidor deve imprimi-la.
- O servidor deve exibir a string rapidamente. Rapidamente significa que, se você acha que demora muito, provavelmente é muito longo.

1 segundo para exibir 100 caracteres é**Demais**!

- Seu servidor deve ser capaz de receber strings de vários clientes seguidos sem precisar reiniciar.
- A comunicação entre seu cliente e seu servidor deve ser feita**apenas**usando sinais UNIX.
- Você só pode usar estes dois sinais:SIGUSR1eSIGUSR2.

O sistema Linux NÃO enfileira sinais quando você já tem sinais pendentes desse tipo! Tempo de bônus?

Capítulo V

Parte bônus

Lista de bônus:

- O servidor reconhece cada mensagem recebida enviando um sinal de volta ao cliente.
- Suporte a caracteres Unicode!

A parte bônus só será avaliada se a parte obrigatória for PERFEITA. Perfeito significa que a peça obrigatória foi executada integralmente e funciona sem avarias. Se você não passou em TODOS os requisitos obrigatórios, sua parte de bônus não será avaliada.

Capítulo VI

Submissão e avaliação por pares

Entregue sua tarefa em seugitrepositório como de costume. Apenas o trabalho dentro do seu repositório será avaliado durante a defesa. Não hesite em verificar novamente os nomes de seus arquivos para garantir que estejam corretos.