

Minitalk

Résumé: Le but de ce projet est de coder un petit programme d'échange de données en utilisant Signaux UNIX.

Contenu

<u>je</u>	<u>Avant-propos</u>	2
<u>II</u>	<u>Instructions communes</u>	3
<u>III</u>	<u>Partie obligatoire</u>	5
<u>IV</u>	<u>Prime</u>	6
<u>V</u>	<u>Soumission et correction par les pairs</u>	7

Chapitre I

Avant-propos

Le cis-3-hexen-1-ol, également connu sous le nom de (Z) -3-hexen-1-ol et alcool de feuille, est une huile incolore liquide avec une intense odeur verte d'herbe d'herbe verte fraîchement coupée et de feuilles.

Il est produit en petites quantités par la plupart des plantes et agit comme un attractif pour de nombreuses insectes prédateurs. Le cis-3-hexén-1-ol est un composé aromatique très important utilisé dans arômes de fruits et de légumes et dans les parfums.

La production annuelle est d'environ 30 tonnes.

Chapitre II

Instructions communes

- Votre projet doit être rédigé conformément à la Norme. Si vous avez un bonus fichiers / fonctions, ils sont inclus dans la vérification de la norme et vous recevrez un 0 s'il y a est une erreur de norme à l'intérieur.
- Vos fonctions ne doivent pas se fermer inopinément (défaut de segmentation, erreur de bus, double gratuit, etc.) en dehors des comportements indéfinis. Si cela se produit, votre projet sera considéré comme non fonctionnel et recevra un 0 lors de l'évaluation.
- Tout l'espace mémoire alloué au tas doit être correctement libéré si nécessaire. Pas de fuites sera toléré.
- Si le sujet l'exige, vous devez soumettre un Makefile qui compilera votre les fichiers source à la sortie requise avec les indicateurs -Wall, -Wextra et -Werror, et votre Makefile ne doit pas se reconnecter.
- Votre Makefile doit au moins contenir les règles \$ (NAME), all, clean, fclean et ré.
- Pour remettre des bonus à votre projet, vous devez inclure un bonus de règle dans votre Makefile, qui ajoutera tous les en-têtes, bibliothèques ou fonctions interdits sur la partie principale du projet. Les bonus doivent être dans un autre fichier _bonus. {C / h}. L'évaluation des parties obligatoires et bonus est effectuée séparément.
- Si votre projet vous permet d'utiliser votre libft, vous devez copier ses sources et ses Makefile associé dans un dossier libft avec son Makefile associé. Tes projets Makefile doit compiler la bibliothèque en utilisant son Makefile, puis compiler le projet.
- Nous vous encourageons à créer des programmes de test pour votre projet même si ce travail **n'aura pas à être soumis et ne sera pas noté** . Ça te donnera une chance pour tester facilement votre travail et celui de vos pairs. Vous trouverez ces tests en particulier

utile lors de votre défense. En effet, lors de la défense, vous êtes libre d'utiliser vos tests et / ou les tests du pair que vous évaluez.

- Soumettez votre travail à votre référentiel git attribué. Seul le travail dans le git repository sera noté. Si Deepthought est assigné à noter votre travail, cela sera fait après vos évaluations par les pairs. Si une erreur se produit dans une section de votre travail pendant La notation de Deepthought, l'évaluation s'arrêtera.
- Les fichiers exécutables doivent être nommés client et serveur.

Minitalk

- Vous devez gérer les erreurs avec sensibilité. En aucun cas, votre programme ne peut se fermer de manière inattendue (Défaut de segmentation, erreur de bus, double libre, etc.).

? Votre programme ne peut pas avoir de fuites de mémoire.

- Sur la partie obligatoire, vous êtes autorisé à utiliser les fonctions suivantes:

- écrire
- signal
- tuer
- getpid
- malloc
- gratuit
- pause
- dormir
- uslepp
- sortie

- JUSTE pour le bonus:

- sigaction

Chapitre III

Partie obligatoire

- Vous devez créer un programme de communication sous la forme d'un client et d'un serveur.
- Le serveur doit être lancé en premier et après avoir été lancé, il doit afficher son PID.
- Le client prendra comme paramètres:
 - Le PID du serveur.
 - La chaîne à envoyer.
- Le client doit communiquer la chaîne passée en paramètre au serveur. Une fois que la chaîne a été reçue, le serveur doit l'afficher.
- La communication entre vos programmes doit être effectuée UNIQUEMENT à l'aide de signaux UNIX.
- Le serveur doit pouvoir afficher la chaîne assez rapidement. Par rapidement, nous entendons que si vous pensez que c'est trop long, alors c'est probablement trop long (indice: 1 seconde pour 100 caractères est COLOSSAL)
- Votre serveur doit pouvoir recevoir des chaînes de plusieurs clients à la suite, sans besoin d'être redémarré.
- Vous ne pouvez utiliser que les deux signaux SIGUSR1 et SIGUSR2.

Chapitre IV

Prime

- Ajouter un petit système d'accusé de réception.
- supporte les caractères Unicode!

Chapitre V

Soumission et correction par les pairs

Soumettez votre travail sur votre référentiel GiT comme d'habitude. Seul le travail sur votre référentiel sera noté.

