Chef de projet : Olivier Descoups

1 :

2 : rendre tout simplement l'écran noir au bout de trois minutes

3.1 : Il sera lancé depuis la ligne de commande d’un terminal GNU/Linux

3.2 :

Type statique : il s’agit d’afficher à l’écran le contenu d’un fichier (avec un format prédéfini) qui représente un dessin.

centré sur la taille de la console

dans la plupart des cas, la taille de la console est de 24 lignes de 80 caractères par ligne

L’écran de veille se débloque et rend la main au shell quand l’utilisateur clique sur n’importe quelle touche du clavier. Sinon, il reste affiché de façon permanente.

Type dynamique : il s’agit d’afficher une information et que, de façon automatique, cette information se mette à jour sans l’intervention de l’utilisateur.

L’information que vous devez afficher est l’heure courante, centrée dans la console, dans le format classique «HH:MM:SS» avec HH de 0 à 23 et MM et SS de 0 à 59.

En bas de la console vous devez faire apparaître un message disant : Cet écran sera actualisé dans quelques secondes……………..

Au bout de N secondes (N doit être paramétrable), réafficher

L’écran de veille se débloque et rend la main au shell quand l’utilisateur tue le processus avec Ctrl^C.

Type Interactif : après un affichage initial, l’écran de veille demande une action à l’utilisateur. En fonction de cette action (clavier), l’affichage doit être modifié. C’est le même principe qu’un jeu interactif mais dans un termSaver.



L’avion sera représenté en 4 positions de la façon suivante (tailles 5x6 ou 6x5). Pour simplifier, il est possible de prendre une seule dimension de 6x6.

L’avion avance selon les commandes de l’utilisateur flèches + entrée et chiffres + entrée

Taille de la console 80x23 car on gardera la dernière ligne pour la saisie des commandes

Sortie de l’écran de veille à tout moment en tapant un caractère bien prédéterminé

Choix d’un des écrans de veille doit se faire de façon automatique et sans l’interaction de l’utilisateur. Pour cela, il est demandé de créer un autre exécutable qui fera la tâche « lanceur » : eXiaSaver.

Fonctionnalité d’historique qui permettra de faire de statistiques sur le nombre de types d’écrans lancés, sur les dates de lancement triées par ordre chronologique ou l’inverse.

Exécutable avec le paramètre « -stats »

3.3.1

Structures de données : Vendredi

I

Le choix de la structure de données est le plus important et doit être fait au début du projet de façon très rigoureuse. De cela dépendront tous les algorithmes que vous allez implémenter par la suite.

3.3.2

Format fichier PBM

Construire un fichier PBM ASCII avec une syntaxe spécifique

Le format de fichier PBM est utilisé pour des images noir et blanc. Un pixel noir est codé par un caractère 1, un pixel blanc est codé par un caractère 0.

Un fichier pbm ASCII a pour nombre magique P1.

Dans les données ascii de l'image, les caractères d'espacement à l'intérieur sont ignorés

Aucune ligne ne doit dépasser 70 caractères.

N’hésitez pas à montrer votre créativité et présenter au jury un jeu de tests conséquent !

Le module de lecture d’un fichier PBM doit être partagé par les 3 exécutables.

3.3.3

Page de code et affichages

Pour les affichages, vous devez utiliser certains caractères ASCII de cette table

Si vous ne trouvez pas les mêmes caractères, vous pouvez les remplacer par d’autres

219

le caractère à afficher du fichier PBM est celui codé par 1 et rien à afficher avec la valeur 0

Pour l’affichage de l’heure pour le niveau dynamique, la taille de chiffres est par défaut de 5 lignes x 3 colonnes

Le séparateur entre les heures, les minutes et les secondes est « : »

Les tailles possibles sont les suivantes : 5x3, 7x4, 9x5, 11x6, etc … jusqu’à un maximum raisonnable pour rentrer tous les chiffres dans une console avec une taille de 80x23.

Stocker que la modélisation des chiffres pour 5x3

3.3.4

Paramétrage et Variables d’environnement

Définir les variables d’environnement

Spécificité d’exiaserveur p11

3.3.5

Historique et statistiques p12

Présentez un menu principal avec 3 ou 4 possibles affichages selon le critère choisi (date, type, nom du fichier, tailles utilisés, … )

3.3.6

Utilisation de primitives et autres fonctions en C

4. Fonctionnalités optionnelles

4.1. Format des images en couleur

Faire le choix de travailler avec des fichiers en couleur

4.2. Gestion de la couleur sous linux

4.3. Gestion du multiprocessus

5. Informations diverses

Tous les développements doivent être réalisés sur un environnement Linux

Vous êtes également invités à installer les « Addons Invité » de VirtualBox

6.

7.

8.

9.

Les livrables attendus sont à rendre le jeudi 15 décembre à 12h00 au plus tard en version électronique sur Git, par mail ou en version papier

Dans la journée du 16 décembre, chaque groupe devra présenter et défendre son travail au cours d’une soutenance