

Rapport de projet

Sommaire

Présentation du groupe.....	2
Reformulation du besoin	2
Contrainte	4
Planification.....	5
Organisation globale du programme.....	6

Présentation du groupe

Le groupe est composé de 2 membres qui sont : Quentin CHAMPAULT et Mandel VAUBOURG.

Reformulation du besoin

Il faut créer un écran de veille pour la console linux.

Le programme eXiaSaver doit être un exécutable qui lancera un "jolie termSaver". Il sera lancer depuis une ligne de commande d'un terminale GNU/Linux.

Il peut être de 3 types différents : Type statique (*On affichera le contenu d'un fichier*), type dynamique (*On affiche une information, qui se mettra à jour sans intervention de l'utilisateur*) ou de type interactif (*Après un affichage initial, l'écran de veille demande une action de l'utilisateur, qui modifiera l'affichage*).

Pour le type statique, l'affichage doit être centré. Pour se débloquent, il faut une action du clavier.

Pour le type dynamique, l'information doit être centré. L'information sera l'heure courante dans le format HH:MM:SS. En bas de la console se trouvera le message « Cet écran sera actualiser dans quelque seconde ». Toute les secondes on ajoutera un point à cette ligne. Le temps d'actualisation doit être paramétrable. Au bout du temps, l'horloge est actualisée et les points « remis à 0 ». Il faut que ctrl+C enlève l'écran de veille.

Pour le type interactif, il faut représenter un avion sur l'écran :
Il a y donc 4 positions de l'avion (6*6). Il faut charger les positions au début du programme. L'avion se déplacera selon les inputs de l'utilisateur (*Z pour haut, S pour bas, Q pour gauche et D pour droite*). S'il n'y a pas d'input, l'avion ira dans la direction du dernier input. On peut aussi déplacer l'avion avec les flèches et les touches du clavier numérique. L'espace aérien est la taille de la console moins la dernière ligne (*Pour la ligne de saisie*) (80*23). Lorsque l'avion atteint le bout de l'écran, il réapparaîtra de l'autre cote de l'écran. Il faut pouvoir voir le bout de l'avion d'un côté alors que la queue est visible de l'autre cote. Quand l'avion tourne, il tourne sur un point central (*le point 3*3 par exemple*). On peut sortir de l'écran de veille en appuyant sur une touche prédéfinie (*E ou X par exemple*).



Le choix d'un écran de veille doit être automatique sans l'interaction de l'utilisateur. Pour faire cela, **il faut un autre exécutable qui fera « lanceur » d'eXiaSaver**.

Il faut que cet exécutable affiche l'historique qui permettra de faire des statistiques sur les types d'écrans lancés. Il faudra lancer le programme avec l'argument « -stats ».

Pour chaque image utilisée dans les écrans de veille (image statique, tous les chiffres de 0 à 9, les avions), on doit construire un fichier PBM ASCII avec la syntaxe suivante à respecter :

Le nombre magique du format :

Il dépend du format et de la variante (binaire ou ASCII).

Un caractère d'espacement (espace, tabulation, nouvelle ligne)

La largeur de l'image (codée en caractères ASCII)

Un caractère d'espacement

La hauteur de l'image (codée en caractères ASCII)

Un caractère d'espacement

Les données binaires de l'image :

L'image est codée ligne par ligne en partant du haut

Chaque ligne est codée de gauche à droite

Toutes les lignes commençant par # sont ignorées.

Un pixel noir sera codé avec un 1 et un pixel blanc avec un 0.

La lecture d'un fichier PBM doit être partagée par les 3 exécutables.

Les variables seront modifiées dans le fichier .profile ou dans le terminal.

Les variables à modifier sont les suivantes :

- EXIASAVER_HOME : répertoire où se trouvent les 3 exécutables. Si cette valeur n'est pas définie, le répertoire par défaut est le répertoire courant.
- EXIASAVER1_PBM : répertoire où se trouvent tous les fichiers PBM pour le niveau statique. Si cette valeur n'est pas définie, le répertoire par défaut est le répertoire courant. Elle sera utilisée par le lanceur pour parcourir tous les fichiers du répertoire et en prendre un de façon aléatoire.
- EXIASAVER2_PBM : répertoire où se trouvent tous les fichiers PBM pour le niveau dynamique. Si cette valeur n'est pas définie, le répertoire par défaut est le répertoire courant.
- EXIASAVER2_TAILLE : taille d'affichage de chiffres dans l'horloge numérique. Si cette variable n'est pas définie, la valeur par défaut est 5x3
- EXIASAVER2_SLEEP : nombre (positif) de secondes entre 2 rafraichissements de l'horloge. Si cette variable n'est pas définie, la valeur par défaut est 10 secondes.
- EXIASAVER3_PBM : répertoire où se trouvent tous les fichiers PBM pour le niveau interactif. Si cette valeur n'est pas définie, le répertoire par défaut est le répertoire courant.

Le lanceur eXiaSaver doit :

- « videra » la console avec l'instruction `system("clear")` ;
- Fera le choix de façon aléatoire entre un des 3 types d'écran
- Lira les variables d'environnement correspondantes au type choisi
- Choisira un fichier image dans le répertoire des images de façon aléatoire si type statique. Prévoyez un jeu de tests conséquent pour mieux voir le côté aléatoire. Utilisez les primitives de parcours d'un répertoire pour connaître son contenu et en choisir un de façon aléatoire.
- Lancera l'exécutable correspondant au niveau choisi avec « exec ou une de ses variantes » en passant les paramètres nécessaires (nom image pour le type écran de veille statique, position initiale de l'avion pour le type interactif).

L'historique stockera :

- La date et l'heure du lancement
- Le type d'écran lancé :
 - o Statique = 1
 - o Dynamique = 2
 - o Interactif = 3
- En fonction du type
 - o Nom du fichier (*Statique*)
 - o Taille de l'horloge (*Dynamique*)
 - o Position initiale de l'avion (*Interactif*)
 - o Durée de actualisation (*Dynamique*)

Pour accéder aux informations de l'historique, on doit créer un menu avec les critères.

Contrainte

La taille des chiffres pour l'heure est de 5*3 pour tous les chiffres.

Le séparateur de HH et MM et de MM et SS est le ':'. **Attention** /\ Il faut adapter l'horloge pour qu'elle soit toujours centrée.

On doit pouvoir paramétrer la taille de l'affichage. Les tailles possibles sont les suivantes :

- 5*3
- 7*4
- 9*5
- 11*6

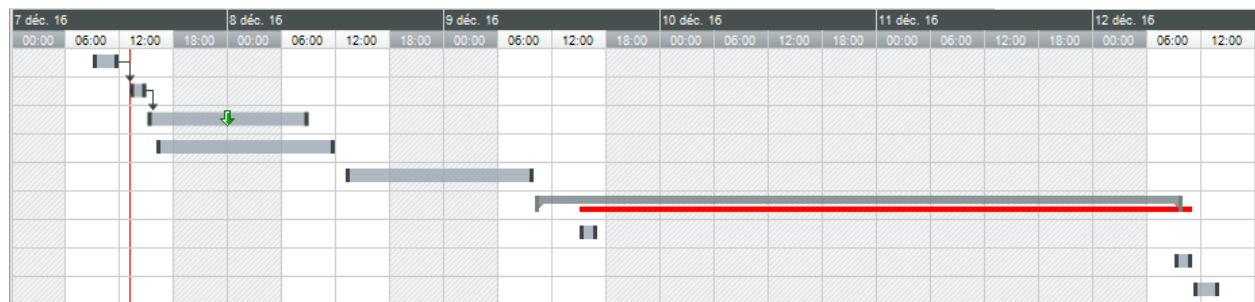
On ne doit stocker que les chiffres en taille 5*3. C'est une contrainte technique imposée. On doit afficher les autres avec des calculs de la modélisation de base.

Voici quelques contraintes techniques :

- Le lanceur doit exécuter un des 3 écrans de veille à travers la commande « exec ou une de ses variantes »
- La lecture des fichiers doit se faire dans un processus fils (fork, getpid, ...)
- Le processus père doit attendre la fin du processus fils pour continuer (wait)
- La manipulation des fichiers devra se faire avec : open, close, fopen, fclose, fgetc, fgets, fprintf ...
- Le parcours d'un répertoire (pour le choix d'une image, choix aléatoire) se fera avec : opendir, readdir, closedir
- La manipulation des chaînes de caractères se fera avec : strcpy, strlen, strcmp, sscanf...
- L'allocation de mémoire dynamique avec malloc et free (tableau des pointeurs par exemple). Intéressant de l'utiliser sachant que les tailles des images PBM sont différentes.

Planification

	Nom de tâche	Durée	Début	Fin
1	Organisation du projet	3 hr	07/12/2016	07/12/2016
2	Installation VM + IDE développement	2 hr	07/12/2016	07/12/2016
3	Ecran de veille statique + modèle	3 hr	07/12/2016	08/12/2016
4	Ecran de veille dynamique	5 hr	07/12/2016	08/12/2016
5	Ecran de veille Interactif	6 hr	08/12/2016	09/12/2016
6	Logiciel de sélection de fond d'éc...	8 hr?	09/12/2016	12/12/2016
7	Historique	2 hr	09/12/2016	09/12/2016
8	Logiciel qui log et lance avec option	2 hr	12/12/2016	12/12/2016
9	Powerpoint + rapport final	2 hr	12/12/2016	12/12/2016



Organisation globale du programme

