**Fonctionnalités requises :**

L’exécutable « termSaver » contient 3 types d’écran de veille différents :

* **Le type statique :**
* Affichage image centré (24x80)
* Reste affiché tant qu’aucune action n’est faite
* **Le type dynamique :**
* Afficher l’heure courante centrée sur la console
* Actualisation de l’heure ("Cet écran sera actualisé dans quelques secondes").
* Chaque seconde ajouter un ".", toutes les N secondes (N paramétré) pour réactualiser l'heure
* EXIT quand l’utilisateur tue le processus (CTRL+C)
* **Le type interactif :**
* Représenter un avion
* Stocker les données dans un fichier PBM
* L’avion s’affiche à un endroit aléatoire sur la console (qui fait 80x23) et dans une des 4 directions, aléatoire aussi (H/B/G/D)
* L’utilisateur doit pouvoir contrôler la direction de l’avion (H/B/G/D + Entrée)
* S’il n’appuie que sur Entrée, l’avion avance avec la dernière commande donnée
* Quand l’avion arrive aux extrêmes, il doit pouvoir ressortir du côté opposé
* Quand il change de sens, il pivote sur son point central
* EXIT de l’écran de veille grâce à une touche prédéfinie
* L’historique :
* Afficher le nombre d’écrans de veille lancés (-stats)
* Afficher Date/Heure/Fichier/Image choisi

Il faut un exécutable pour faire un random entre les 3 écrans de veille.

**GRILLE D'EVALUATION :**

Obligatoires/Principaux :

* **Le lanceur :**
* Il existe et il peut être lancé depuis la console sans paramètres
* Le lanceur "vide" la console lors de l'exécution
* Historique - Enregistre les informations dans le fichier "historique"
* Le programme interprète correctement le paramètre '-stat' et ne lance pas un écran de veille
* Affichage menu stat avec plusieurs choix
* Calcul et affichage statistiques
* Choisit aléatoirement un de trois écrans
* Il lance bien un des 3 exécutables avec les bons paramètres
* Accès aux variables d'environnement et lecture de la valeur
* Parcours répertoire des images PBM (ou PPM)
* Choix aléatoire d'une image pour le lanceur de type statique
* Jeu de test pour les images - au moins 5 images de tailles différentes
* On voit l'implémentation d'un algorithme de tri simple pour les stats
* Variables d'environnement définies - à la main ou dans le .profile

Deuxième temps :

**exiaSaver 1 - Type statique :**

* Image chargée correctement
* Affichage image centrée dans la console
* Déblocage en touchant sur n'importe quelle touche
* Utilisation de fork pour chargement de l'image

**exiaSaver 2 - Type dynamique :**

* Images de chiffres chargées en mémoire (normalement 1 seule fois)
* Affichage heure centrée dans la console
* Affichage message d'attente et réactualisation
* On voit bien passer les secondes (10" par défaut) entre 2 réaffichages
* Affichage heure centrée dans la console après refresh auto. de l'écran
* Déblocage du processus par Ctrl+C
* Affichage heure avec tailles différentes (5x3, 7x4, …) Au moins 2 !

**exiaSaver 3 - Type interactif :**

* Images des avions chargées en mémoire depuis les fichiers PBM
* Affichage de l'avion à la position (initiale) passée en paramètre & choix aléatoire de la direction et du sens.
* L'avion réagit quand on tape sur une des 4 touches de commande
* Affichage de l'avion à sa nouvelle position
* Si on ne tape rien (seulement Entrée) , l'avion avance dans le même sens, d'une case à la fois
* Quand l'avion sort de l'espace aérien, il rentre du côté opposé selon spécification
* Sortir de l'écran en tapant une touche précise

**Environnement :**

* Linux et Git opérationnels

**Technique :**

* Utilise structures dynamiques (tableaux des pointeurs, listes chaînées) ou des structures statiques
* Lecture et écriture des fichiers fonctionnent parfaitement (images PBM, historique, …)
* Utilisation correcte des primitives/fonctions demandées (et/ou d'autres)
* Le code est complétement commenté
* Chaque exécutable est découpé au moins en 3 fichiers & réutilisation du code de lecture des fichiers entre les 3 exécutables exiaSaver

**Divers :**

* Tous les livrables ont été réalisés et ils sont de qualité.

**Bonus groupe**(Fonctionnalités optionnelles) :

* Utilisation des fichiers PBM ou équivalent
* Utilisation des couleurs
* Utilisation de plusieurs forks en "parallèle" pour la lecture des fichiers et/ou calculs affichage et/ou la capture touches écran de type interactif