

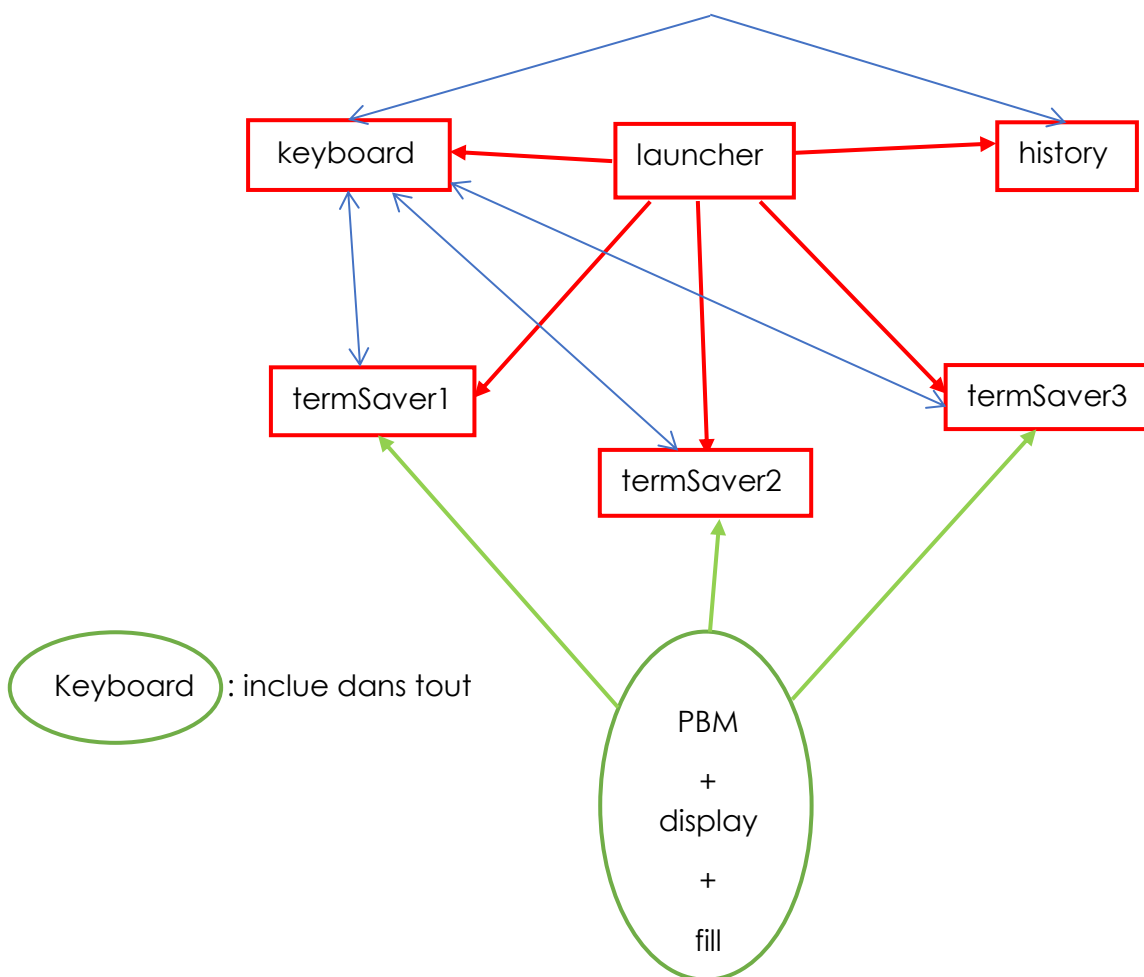
: programme

: .h

→ : exec

→ : communication par Pipe

→ : include



Structures personnalisées :

background

```
// contient l'image à afficher
dans la console
//en pixel, hauteur totale de la
console
int height
//largeur totale
int width
//tableau contenant l'image à
afficher, 1 pixel est une case
de tableau, valant 0 ou 1
int grid (tableau dynamique
double entrée)
//utilisé dans termSaver3
struct plane
```

plane

```
//contient les informations
et l'image de l'avion
//direction vers laquelle il
se dirige, codée avec un int
int direction
//position en hauteur
int xpos
//position en largeur
int ypos
//image de l'avion (dépend
de la direction)
int plane (tableau 2 entrée
statique 6*6)
```

dynamic

```
//stockage des images pbm
des 9 chiffres
int number0 (tableau 2
entrée statique)
int number1 (idem)
...
int number9
//stockage du message en
pbm « refresh in... »
int message (tableau 2
entrée statique)
```

history

```
//Liste chaînée où l'on stocke
l'historique complet chargé depuis un
fichier externe
Struct history {
    //tm inclut dans time.h permet de
stocker la date et l'heure
    Struct tm instant
    //quel fond a était affiché
    int screenlaunched
    //en pixel, hauteur totale de la
console
    int height
    //largeur totale
    int width
    //position en hauteur (de l'avion)
    int xpos
    //position en largeur
    int ypos
}
```

image

```
//stocke l'image pbm du
fond, fonctionne pareil que
grid
int image (tableau
dynamique 2 entrée)
```

launcher.c

```
//lanceur lancé depuis le terminal Linux
//random utilisé pour le choix du fond d'écran a lancer, en fonction du nbre d'images dispos
int random(int)
//rentre dans la structure tm la date et heure actuelle
tm getHour(tm)
//sauvegarde dans un fichier externe d'historique les informations du lancement actuel
void fillHistory(tm, choice)
//lance le fond choisi et keyboard, ou le programme history suivant le paramètre
void launchProg(choice)
```

termSaver1.c

```
//Ecran de veille statique
//initialise les variables de background (sauf avion car non requis ici)
background initBackground (background)
//fait un random entre 0 et le nombre d'images disponibles pour choisir quel fond afficher
int imageChoice(int)
//initialise l'image choisie
image fillImage(image)
//intègre l'image au background
background fillBackground(background, image)
//affiche le fond (background)
void display(background)
//communique avec keyboard.c pour savoir si espace est tapé, et si oui tue le processus
int getKey()
```

termSaver2.c

```
//Ecran de veille dynamique
//initialise la struct dynamic
dynamic fillDynamic(dynamic)
//initialise les variables de background (sauf avion car non requis ici)
background initBackground(background)
//rentre dans la structure tm la date et heure actuelle
tm getHour(tm)
//intègre dans le background les images statiques, comme les :)
background fillStatics(background)
//intègre dans le background l'heure
background fillBackground2(background,dynamic,tm)
//affiche l'image
void display(background)
//communique avec keyboard.c pour savoir si espace est tapé, et si oui tue le processus
int getKey()
```

termSaver3.c

```
//Ecran de veille interactif
//initialise les variables de background
background initBackground (background)
//initialise aléatoirement la position et direction de l'avion
plane initPlane(plane, background)
//remplie l'image de l'avion, suivant la direction
plane fillPlane(plane)
//communique avec keyboard.c pour savoir si espace est tapé, et si oui tue le processus, et
renvoie ici un int correspondant à une flèche si elle est tapée
int getKey()
//change la direction de l'avion si une flèche est renvoyée par getKey
plane changeDirection(int)
//fait varier les variables positions de l'avion en fonction de la direction
plane movePlane(plane)
//intègre l'image de l'avion au background au bon endroit et direction
background fillBackground3 (background, image)
//affiche le fond (background)
void displayBackground(background)
```

history.c

```
//Permet de lire l'historique
//depuis le fichier de sauvegarde, remplit la liste chaînée
history fillHistory(history)
//propose de faire un tri de la liste
history sortHistory(history)
//affiche l'historique
void displayHistory(history)
```

PBM.h

```
//permet de placer dans la
structure correspondante l'image
voulue, pour la placer ensuite dans
le background qu'on affichera
Struct image
Struct dynamic
Struct plane
Image fillImage(image)
Dynamic fillDynamic(dynamic)
Plane fillPlane(plane)
```

fill.h

```
//intègre les images envoyées dans le
background aux bonnes coordonnées
background fillBackground1 (background,
image)
background fillBackground2 (background,
dynamic)
background fillBackground3 (background,
plane)
```

display.h

```
//affiche le fond à l'écran à partir
du tableau grid
displayBackground(background)
```

keyboard.h

```
//si on appuie sur une touche
voulue (espace ou une des 4
flèches pour l'avion) renvoie le int
correspondant par pipe
int getKey()
```