#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

//Bibliothèque contenant les fonctions pour le parcours de dossiers

#include <dirent.h>

// Si le système n'est pas windows, alors inclure sys/types.h

#ifndef WIN32

#include <sys/types.h>

#endif

//On inclut les prototypes des fonctions du launcher

#include "launcher.h"

int main(int argc,char \*argv[])

{

system("clear");

srand(time(NULL));

int choix\_ts=nb\_rand()%3+1;

// int nb\_I\_S;

// nb\_I\_S = nombreImagesStatique();

int choix\_I\_S=nb\_rand()%5+1;

printf(" -------------------------------\n");

printf("| Bienvenue dans ExiaSaver |\n");

printf(" -------------------------------\n");

sleep(3);

printf("\nNous allons voir maintenant quel écran de veille vas se lancer\n\n\n");

sleep(5);

printf("L'écran de veille choisi est le numéro %d\n\n", choix\_ts);

if (choix\_ts ==1)

{

printf("Nous allons lancer le type statique !\n");

//L'IMAGE QUE L'ON VA LANCER

/\*switch (nbPic)

{

case 1:

break;

case 2:

break;

case 3:

break;

case 4:

break;

}\*/

}

else if (choix\_ts ==2)

{

printf("Nous allons lancer le type dynamique !\n");

}

else if (choix\_ts ==3)

{

printf("Nous allons lancer le type interactif !\n");

}

saveDate(choix\_ts,choix\_I\_S);

}

//FONCTION NOMBRE RANDOM

int nb\_rand(void)

{

return (rand());

}

//FONCTION POUR EXPORTER DANS L'HISTORIQUE

void saveDate(int type, int nombrePicture)

{

FILE\* fhistorique = NULL; //initialise le pointeur à NULL

fhistorique = fopen("historique.txt", "r+"); //OUVRE LE FICHIER historique.c

if (fhistorique != NULL) //Vérifie si le fichier est bien ouvert, car si fichier=NULL signifie qu'il y a une erreur

{

printf("\nSuccès pour l'ouverture de l'historique\n");

//RECUPERER L'HEURE ET LA DATE

//Lire l'heure courante

time\_t HeureDate;

HeureDate = time(NULL);

//Convertir l'heure en heure locale

struct tm tm\_HeureDate = \*localtime (&HeureDate);

//Crée une chapine JJ/MM/AAAA HH:MM:SS

char s\_HeureDate[sizeof "JJ/MM/AAAA HH:MM:SS"];

strftime (s\_HeureDate, sizeof s\_HeureDate, "%d/%m/%Y %H:%M:%S", &tm\_HeureDate);

fseek(fhistorique, 0, SEEK\_END); //Place le curseur à la fin du fichier

//Récupère le paramètre associé

//ECRITURE DE TOUTES LES DONNEES (SAUF PARAMETRE) DANS L'HISTORIQUE

fseek(fhistorique, 0, SEEK\_END); //Place le curseur à la fin du fichier

//Ecrit la date

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"%s", s\_HeureDate);

fprintf(fhistorique," | ");

//Ecrit le type lancé

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique, "%d", type);

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique," | ");

//ECRITURE DES PARAMETRES ASSOCIES AU TERMSAVER CHOISI

if(type == 1)

{

//Ecrit l'image utilisé pour le type statique

switch (nombrePicture)

{

case 1:

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"test.pbm\n");

break;

case 2:

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"test.pbm\n");

break;

case 3:

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"test.pbm\n");

break;

case 4:

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"test.pbm\n");

break;

case 5:

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"test.pbm\n");

break;

}

}

else if (type == 2)

{

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"\n");

}

else if (type == 3)

{

fseek(fhistorique, 0, SEEK\_END);

fprintf(fhistorique,"\n");

}

fclose(fhistorique);//FERME LE FICHIER

printf("Ecriture dans l'historique réussie\n\n");

}

else

{

printf("\nImpossible d'ouvrir le fichier historique...");

}

}

// //FONCTION POUR CONNAITRE LE NOMBRE D'IMAGE

// int nombreImagesStatique(void)

// {

// int nombrePic = 0; //initalisation à 0 du nombre d'images

// DIR\* repertoireImages = NULL; //déclaration et initialisation à NULL du pointeur repertoireImages

// struct dirent\* ent = NULL;

// repertoireImages = opendir("/Avion"); //Ouverture du dossier PBM\_statique dans repertoireImages

// while ((ent = readdir(repertoireImages)) != NULL) //Tant que des fichiers sont lus (fichiers déjà tous lu = NULL)

// {

// if (strcmp(ent->d\_name, ".") != 0 && //Si le fichier lu n'est pas .

// strcmp(ent->d\_name, "..") != 0) //Et n'est pas .. non plus

// nombrePic++; //Alors on incrémente le compteur

// }

// closedir(repertoireImages);

// return nombrePic;

// }