Ecole Nationale Supérieure des Mines
de Rabat
2020/2021

Rapport de Projet ASE

Simulation d'une gestion de la mémoire par pagination

Réalisé par:

FARAH Manal
ATMANI Houda
FARSSI Rahma
HANDI Kaoutar
MAKHCHOUN Khadija
OUZOUGAGH Chaimaa

Encadré par:

Mme.OUAZZANI Khadija

Table des matières

Introduction

La gestion de mémoire est l'une des composantes du système d'exploitation. Il a pour rôle de réserver et de gérer les données sur la mémoire centrale en suivant les parties de la mémoire qui sont utilisées ou non. Ainsi, en allouant et libérant l'espace mémoire aux programmes.

En effet, un gestionnaire de la mémoire doit assurer une meilleure exploitation de la mémoire centrale. De plus, il est très intéressant de résoudre le problème de la fragmentation, et d'augmenter le niveau de la multiprogrammation. A priori il y a plusieurs stratégies pour la gestion, mais, nous allons s'intéresser précisément à la gestion par **pagination** qui répond parfaitement à ces exigences.

La pagination consiste à répartir un programme en des pages de taille égale. La même façon de répartition est faite pour la mémoire centrale afin d'assurer l'implémentation du programme dans celle-ci d'une manière non contiguë en gardant la trace des parties utilisés et des parties libres dans la mémoire et en plaçant chacune des pages dans les premières pages libérées.

Dans ce qui suit, on présentera tout d'abord le cadre du projet en traitant son cahier des charges fonctionnel, ensuite on va spécifier les algorithmes des fonctions principales du système pour aboutir enfin à la réalisation concrète de ce programme à travers des jeux d'essai.

1. L’objectif du projet

La gestion de la mémoire par pagination désigne l’organisation de l’ensemble des programmes à savoir : le chargement, affichage de l’espace vide et l’espace occupé par ces programmes, la suppression, la recherche d’un programme.

Un tel programme nous permet de comprendre les fondements de la gestion mémoire en utilisant les concepts de pagination ainsi que leurs avantages comparés.

Dans ce contexte s’inscrit le but de notre projet à savoir la création d’une application qui fait la simulation de la gestion de la mémoire par pagination.

2. Description des fonctionnalités du programmes

Le système propose un menu dont les fonctionnalités attendues sont :

- Charger un programme : création d’un nouveau programme définit par son nom et sa taille en registre et par sa liste des pages.
- Afficher les pages vides
- Afficher la lise des programmes
- Chercher un programme
- Afficher les programmes (avec affichage des pages occupées pour chaque programme)
- Terminer l'exécution d'un programme
- Afficher la mémoire centrale (état de mémoire)
- Vider la mémoire : supprimer tous les programmes chargés
- Quitter : sortir du programme et perdre l’état de la mémoire

3. Les structures de données utilisées

L'implémentation est basée sur l'utilisation des listes chaînées. Une liste chainée est une structure de données dynamique qui représente un ensemble d'éléments de même taille et de même type. Chaque élément est défini par des variables et nécessairement une variable de type pointeur qui pointe vers l'élément suivant et le dernier élément a un pointeur nul

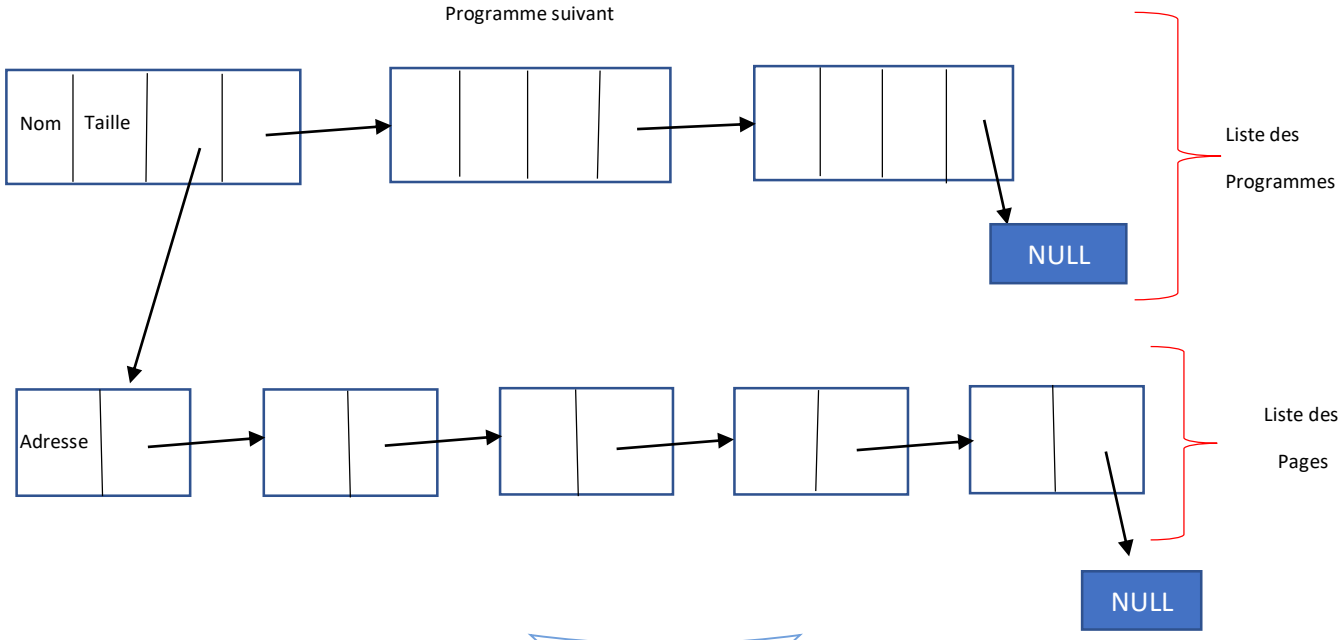
Dans notre projet, on a utilisé deux listes chainées :

1) Une liste des pages libres contenant deux champs

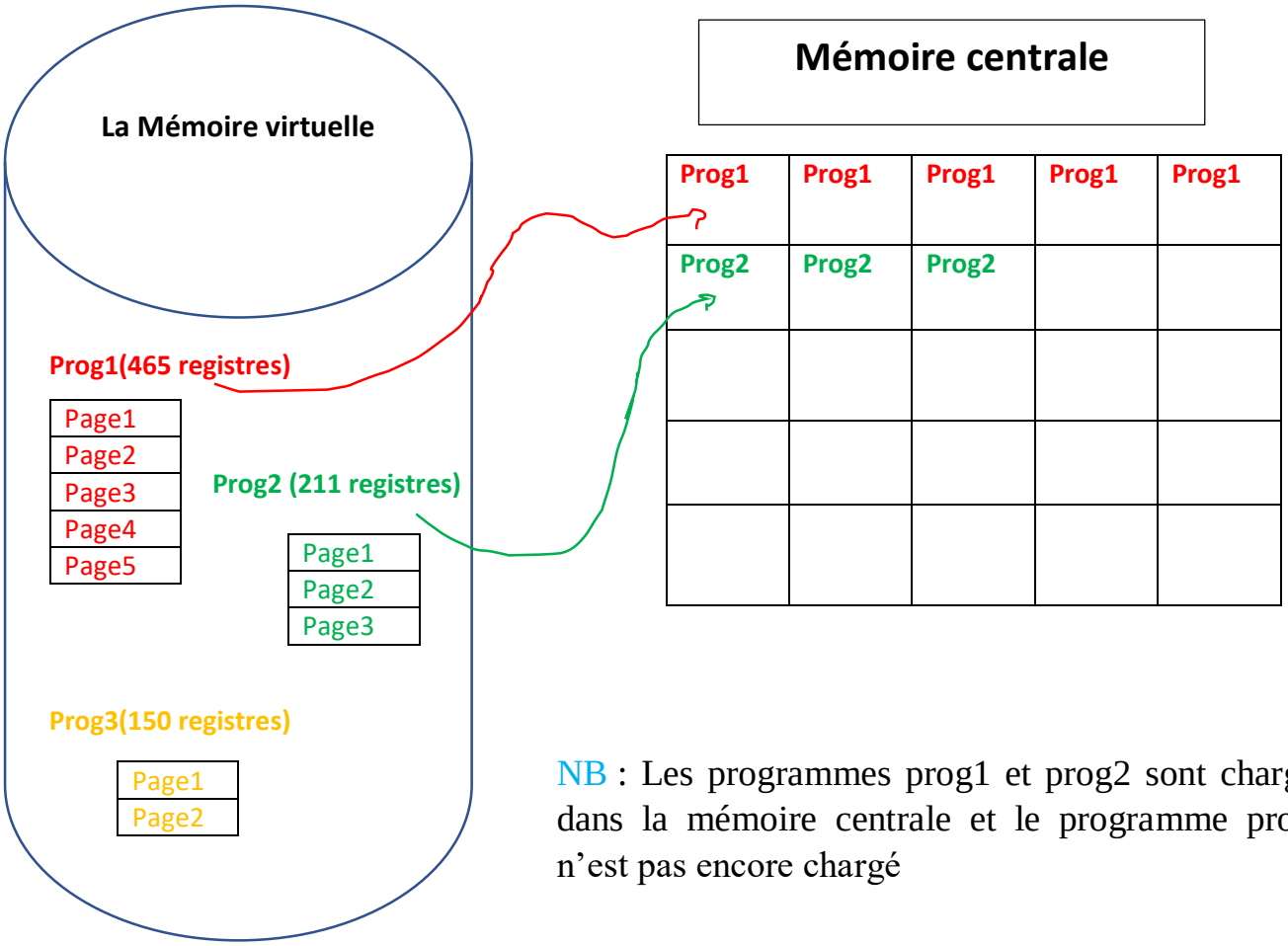
- adresse : représente l'adresse de la page vide
- Page_svt : est un pointeur qui pointe vers la page libre suivante

2) Liste des programmes représente une liste des listes (chaque programme contient une liste des pages) et contenant quatre champs :

- nom : le nom du programme
- taille : la taille du programme donnée en nombre d'instructions tel que chaque cent instruction représente une page occupée par un programme.
- page_prog : est un pointeur pointant vers la liste des pages occupées par le programme
- prog_suiv : est un pointeur pointant vers le programme suivant



On considère aussi que la mémoire centrale contient 25 pages et chaque page = 100 registres.



NB : Les programmes prog1 et prog2 sont chargés dans la mémoire centrale et le programme prog3 n'est pas encore chargé

4. Le langage utilisé

L'implémentation du programme s'est appuyé sur le **langage C** qui est un langage de programmation impératif conçu pour la programmation système, tout en le compilant à l'aide de l'**IDE** (Environnement de développement intégré) **Code::Blocks**.

5. Les algorithmes des fonctions principales

a) Déclaration des structures

main.c main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define cl "\x1b[35m"
#define cs "\x1b[34m"
#define defaut   "\x1b[30m"

//structures utilisées


//liste des pages
struct page{
    int adresse;
    struct page *page_svt;
};

//liste programme
struct prog{
    char nom[15];
    int taille;
    struct page *page_prog;
    struct prog *prog_svt;
};
```

b) Code main

```c
int main()
{
   // system("color 30");
    char choix[15];
    int t;
    char n[15];
    int taille;
    struct page * p = NULL;
    struct page *pg;
    struct prog *pgm=NULL;
    p=(struct page*)(malloc(sizeof(struct page)));
    p->adresse=0;
    p->page_svt=NULL;
    p=creation_liste_espace(p);


/*Afficher le menu à l'utilisateur*/
    system("cls");
    system("cls");
    printf("\n\n");
    printf("\n\t\t\t\t\t\t\t    Bonjour (^_^) !!\n\t\t\t\t\t\t   Gestion de la memoire par pagination\n ");
    printf("\n                              +-----------------Ce programme a ete realise par:---------------+\n");
    printf("                              |                    * HANDI Kaoutar                             |\n");
    printf("                              |                    * MAKHCHOUN Khadija                         |\n");
    printf("                              |                    * ATMANI Houda                              |\n");
    printf("                              |                    * FARAH Manal                               |\n");
    printf("                              |                    * OUZOUGAGH Chaimae                         |\n");
```

```
77      printf("                          |                    * FARSSI Rahma                          |\n");
78      printf("                          *-------------------------Encadre par :----------------------*\n");
79      printf("                          |                 Mme.TOUHAMI OUAZZANI Khadija               |\n");
80      printf("                          *--------------------------------------------------------------*\n\n\n");
81
82
83
84  while (strcmp(choix,"9")!=0) {
85
86      printf("\n                                +-----------------------------------------------------+\n");
87      printf("                              |                      ~~ MENU ~~                     |\n");
88      printf("                              +-----------------------------------------------------+\n");
89      printf("                              |         1 . Charger un programme                    |\n");
90      printf("                              |         2 . Afficher les pages vides                |\n");
91      printf("                              |         3 . Afficher la liste des programmes        |\n");
92      printf("                              |         4 . chercher un programme                   |\n");
93      printf("                              |         5 . Afficher les programmes par pages       |\n");
94      printf("                              |         6 . Terminer l'execution d'un programme     |\n");
95      printf("                              |         7 . Afficher la memoire centrale            |\n");
96      printf("                              |         8 . Vider la memoire                        |\n");
97      printf("                              |         9 . Quitter                                 |\n");
98      printf("                              *-----------------------------------------------------*\n");
99
100         printf("\n--------> Veuillez selectionner une option: ");
101         scanf("%s", &choix);
102         printf("\n");
```

```
103     if ( strcmp(choix,"1")==0) {
104
105         printf("--------> Veuillez saisir le nom du programme a charger: ");
106         scanf("%s", &n);
107         printf("--------> veuillez saisir la taille de ce programme: ");
108         scanf("%d", &taille);
109
110         pgm=charger_prgm(pgm,n, taille, p);
111     }
112     else if (strcmp(choix,"2")==0) {
113         int l=Nbr_Elemt(p);
114      if(l==0){
115         printf("\n----------Aucune page vide n'existe dans la memoire !!!! \n\n");
116      }
117      else{
118        printf("\n----------Les pages vides dans la memoire : \n\n");
119         afficher_espace(p);
120         printf("\n");
121      }
122     }
123     else if (strcmp(choix,"3")==0) {
124
125         int s=Nbr_Elemt(p);
126         if(s==25){
127             printf("\n");
128             printf("----------la memoire est vide !!!");
129             printf("\n");
130         }
```

```
131         else{
132             printf ("\n----------La liste des programmes : \n\n");
133             afficher_prog(pgm);
134             printf("\n");
135         }
136     }
137     else if (strcmp(choix,"4")==0) {
138         printf("--------> Veuillez saisir le nom du programme a afficher: ");
139         scanf("%s", &n);
140         int v=charcher(pgm,n);
141        if(v==0){
142             printf("\n");
143          printf("----------Ce programme n'existe pas !!!!\n");
144
145        }
146        else{
147           printf("\n");
148           printf("\n----------Les pages du programme %s : \n\n",n);
149           printf("\t    ");
150           afficher_pages_prog(pgm,n);
151           printf("\n");
152        }
153
154     }
155     else if (strcmp(choix,"5")==0) {
156         int s=Nbr_Elemt(p);
157             if(s==25){
158             printf("\n");
```

```
156             int s=Nbr_Elemt(p);
157                 if(s==25){
158                 printf("\n");
159                 printf("--------!!!la memoire est vide!!!--------");
160                 printf("\n");
161             }
162             else{
163                printf("\n----------La Liste des programmes : \n\n");
164                afficher_prog_pages(pgm);
165             }
166
167         }
168         else if (strcmp(choix,"6")==0) {
169
170             printf("--------> Veuillez saisir le nom du programme a supprimer: ");
171             scanf("%s", &n);
172            int v=charcher(pgm,n);
173           if(v==0){
174              printf("----------Ce programme n'existe pas !!!!!\n");
175           }
176           else{
177             pg=get_listpage(pgm,n);
178             pg=supprimerElementEnFin(pg);
179             pgm=supprimer_prog(pgm, n);
180             printf("\n--------le programme a ete bien supprime-------\n");
181             p=fusionner(p->page_svt,pg);
182           }
183         }
```

```
182           }
183         }
184         else if (strcmp(choix,"7")==0) {
185         etat_memoire(pgm);
186         }
187         else if(strcmp(choix,"8")==0){
188             if(pgm==NULL){
189                 printf("\n----------La m%cmoire est d%cia vide !!!!",130,138);
190             }
191             else{
192                 while(pgm!=NULL){
193                 pg=get_listpage(pgm,pgm->nom);
194                 pgm=supprimer_prog(pgm, pgm->nom);
195                 p=fusionner(p->page_svt,pg);
196             }
197             printf("----------La m%cmoire a bien %ct%c vid%c ",130,130,130,130);
198             }
199         }
200         else {
201
202             printf(" !!!!!!!! veuillez saisir un numero depuis le menu !!!!!!! \n");
203
204         }
205 }
206  printf("\n\n *~~~~~~~~~~ Merci pour avoir utilise ce programme (^_^) a tres bientot ~~~~~~~~~~*  \n\n");
207 return 0;
208 }
```

c) La création des pages vides

Création et initialisation des pages libres

```
210 struct page * creation_liste_espace(struct page *p){
211         struct page* nouveau = NULL;
212         nouveau=(struct page*)(malloc(sizeof(struct page)));
213         nouveau->adresse=1;
214         nouveau->page_svt=NULL;
215         p->page_svt=nouveau;
216     for (int i =2; i<=26; i++){
217         struct page * courant = NULL;
218         courant=(struct page*)(malloc(sizeof(struct page)));
219         courant->adresse=i;
220         courant->page_svt=NULL;
221         nouveau->page_svt=courant;
222         nouveau= nouveau->page_svt;
223     }
224   return p;
225 }
226
```

d) Chargement d'un programme

Avant de charger un programme il faut créer la liste des pages de ce dernier en fonction du taille

La création des pages du programme.

```
254  struct page * creation_page(struct page *p, int t){
255    struct page * pg = NULL;
256      pg=(struct page*)(malloc(sizeof(struct page)));
257
258      pg->adresse=0;
259      pg->page_svt=NULL;
260      struct page * nouveau = NULL;
261      nouveau=(struct page*)(malloc(sizeof(struct page)));
262      nouveau->adresse=p->page_svt->adresse;
263      nouveau->page_svt=NULL;
264      pg->page_svt=nouveau;
265
266      for(int i=2; i<=t+1; i++){
267          struct page * courant = NULL;
268          courant=(struct page*)(malloc(sizeof(struct page)));
269          courant->adresse=p->page_svt->page_svt->adresse;
270          courant->page_svt=NULL;
271          nouveau->page_svt=courant;
272          nouveau= nouveau->page_svt;
273          p=p->page_svt;
274
275          }
276     return pg;
277  }
```

Supprimer la liste des pages du programme depuis la liste des pages vides.

```
291  struct page * supprimer_pages_charges(struct page *pr,int taille){
292          struct page *nouveau;
293          for (int i=0; i<taille ;i++){
294           nouveau=pr->page_svt;
295           pr->page_svt=pr->page_svt->page_svt;
296            free(nouveau);
297           }
298     return pr;
299  }
300
```

L'ajout du programme à la liste des programmes.

```
324  struct prog* charger_prgm(struct prog * pgm,char l[],int taille,struct page *p){
325    if(taille<=0){
326       printf("\n----------!!!taille insuportable!!!! \n");
327    }
328    else{
329         struct page * nv = NULL;
330         int t=trouver_nbrPages(taille);
331         int n=Nbr_Elemt(p);
332         int v=charcher(pgm,l);
333    if(v==0){
334        if (t>n){
335           printf("\n----------!!!taille insuportable!!! \n");
336        }
337      else{
338            struct page * Pg = NULL;
339            Pg=(struct page*)(malloc(sizeof(struct page)));
340            Pg=creation_page(p, t);
341            struct page * nv = NULL;
342           nv=(struct page*)(malloc(sizeof(struct page)));
343           nv=supprimer_pages_charges(p,t);
344            printf("\n");
345           if (pgm==NULL){
346              pgm=creation_programme(Pg,NULL,l,t);
347              printf("\n--------------le programme a ete bien enregistre \n");
348           }
349           else{
350              struct prog *tmp;
351              tmp=(struct prog*)(malloc(sizeof(struct prog)));
352              tmp=pgm;
353              while(tmp->prog_svt != NULL){
354
355                   tmp = tmp->prog_svt;
356              }
357             tmp->prog_svt = creation_programme(Pg,NULL,l,t);
358             printf("\n--------------le programme a ete bien enregistre \n");
359           }
360         }
361    }
362    else{
363      printf("\n----------!!!!ce programme existe deja!!!! ");
364    }
365    }
366   return pgm;
367  }
```

e) Afficher les pages vides

```
227  void afficher_espace(struct page* p){
228
229      while(p->page_svt->page_svt!=NULL){
230          printf(" %d  ", p->page_svt->adresse);
231          p=p->page_svt;
232      }
233  }
```

f) Afficher les programmes par pages

```
298  void afficher_prog(struct prog *p){
299      while(p!=NULL){
300          printf("-----------------# %s \n",p->nom);
301          p=p->prog_svt;
302      }
303  }
304
305  void afficher_prog_pages(struct prog *p){
306      while(p!=NULL){
307          printf("-----------------# %s  :",p->nom);
308          afficher_page(p->page_prog);
309          printf("\n\n");
310          p=p->prog_svt;
311      }
312  }
```

g) Chercher un programme

```
369  void afficher_pages_prog(struct prog *p,char l[]){
370
371      while(p!=NULL){
372
373          if(strcmp(l,p->nom)==0){
374          afficher_page(p->page_prog);
375           }
376          p=p->prog_svt;
377      }
378  }
```

h) Terminer l'exécution d'un programme

Pour terminer l'exécution on doit tout d'abord récupérer les pages du programme.

```
585  struct page* get_listpage(struct prog *p,char l[]){
586
587    struct prog * tmp = (struct prog * ) malloc(sizeof(struct prog * ));
588    tmp = p;
589    while (tmp) {
590      if (strcmp(tmp->nom, l) == 0) {
591        return tmp->page_prog;
592      }
593      tmp = tmp->prog_svt;
594    }
595  }
```

Et après on peut l'insérer dans la liste des pages vides à l'aide d'une fonction de fusion.

```
614  struct page* fusionner(struct page* p,struct page* pg)
615  {
616      struct page* temp1=p;
617      struct page* temp2=pg;
618
619        if(temp1==NULL)
620        {
621           return temp2;
622        }
623        else if(temp2==NULL)
624        {
625           return temp1;
626        }
627        else if ((temp1->adresse) <= (temp2->adresse))
628       {
629            temp1->page_svt = fusionner(temp1->page_svt,temp2);
630            return(temp1);
631       }
632        else
633        {
634            temp2->page_svt = fusionner(temp1,temp2->page_svt);
635            return(temp2);
636       }
637  }
```

Et finalement on supprime le programme.

```
404  struct prog * supprimer_prog(struct prog *pgm, char l[]){
405      if(pgm == NULL)
406          return NULL;
407      if(strcmp(l,pgm->nom)==0)
408      {
409          struct prog* tmp = pgm->prog_svt;
410          free(pgm);
411          tmp = supprimer_prog(tmp, l);
412          return tmp;
413      }
414      else
415      {
416          pgm->prog_svt = supprimer_prog(pgm->prog_svt,l);
417          return pgm;
418      }
419  }
```

i) Afficher la mémoire centrale

```
469  /*************************************Affichage*************************************************************/
470
471  char*  etat_memoire(struct prog *listeProg){
472
473      struct prog *p;
474      p=listeProg;
475
476      char T[25][10]={"","","","","","","","","","","","","","","","","","","","","","","","",""};
477
478      while(p!=NULL){
479               struct page *temp;
480                temp=p->page_prog;
481                while(temp->page_svt->page_svt!=NULL){
482
483                temp=temp->page_svt;
484
485                int j=temp->adresse;
486                strcpy(T[j-1],p->nom);
487            }
488           temp=NULL;
489
490           p=p->prog_svt;
491      }
492
493     printf(c5);
494     printf("\n\t+*************************************************************************+\n\t");
495     printf("|\t\t\t\t\t\t\t\t\t     |\n");
```

```
497     printf(c1);
498     printf("Memoire Centrale");
499     printf(defaut);
500     printf(c5);
501     printf("   \t\t\t     |\n");
502     printf("\t|\t\t\t\t\t\t\t\t     |\n");
503     printf("\t+************++************++************++************++***********+\n\t");
504     printf(defaut);
505      for (int i=0;i<5;i++){
506          formater(T[i]);
507      }
508      printf("\n");
509      printf(c5);
510      printf("\t+************++************++************++************++************+\n\t");
511      printf(defaut);
512      for (int i=5;i<10;i++){
513          formater(T[i]);
514      }
515      printf("\n");
516      printf(c5);
517      printf("\t+************++************++************++************++************+\n\t");
518      printf(defaut);
519      for (int i=10;i<15;i++){
520          formater(T[i]);
521      }
522      printf("\n");
523      printf(c5);
524     printf("\t+************++************++************++************++************+\n\t");
```

```
516       printf(c5);
517       printf("\t+************++************++************++************++***********+\n\t");
518       printf(defaut);
519       for (int i=10;i<15;i++){
520           formater(T[i]);
521       }
522       printf("\n");
523       printf(c5);
524      printf("\t+************++************++************++************++************+\n\t");
525      printf(defaut);
526       for (int i=15;i<20;i++){
527           formater(T[i]);
528       }
529       printf("\n");
530       printf(c5);
531       printf("\t+************++************++************++************++***********+\n\t");
532       printf(defaut);
533       for (int i=20;i<25;i++){
534           formater(T[i]);
535       }
536       printf("\n");
537       printf(c5);
538       printf("\t+************++************++************++************++***********+\n\n\t");
539       printf(defaut);
540       return T;
541  }
```

j) Vider la mémoire

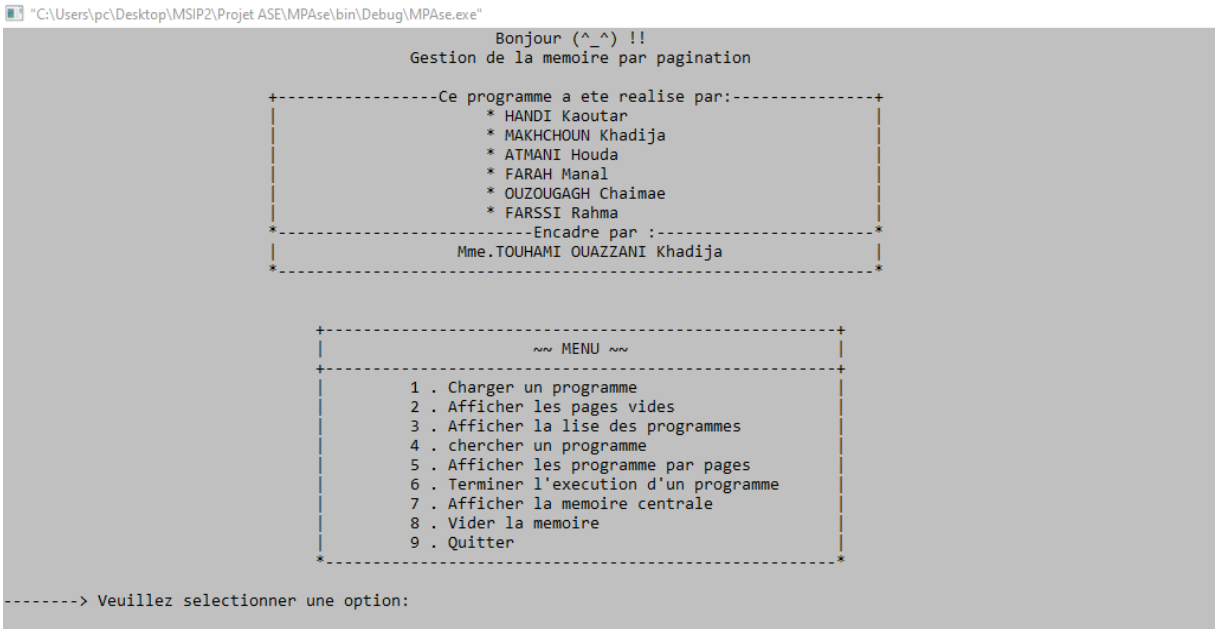
Permet de supprimer tous les programmes.

```
192             if(pgm==NULL){
193                 printf("\n----------La m%cmoire est d%cja vide !!!!",130,138);
194             }
195             else{
196                 while(pgm!=NULL){
197                 pg=get_listpage(pgm,pgm->nom);
198                 pg=supprimerElementEnFin(pg);
199                 pgm=supprimer_prog(pgm, pgm->nom);
200                 p=fusionner(p->page_svt,pg);
201             }
202             printf("----------La m%cmoire a bien %ct%c vid%c ",130,130,130,130);
203             }
204         }
```

6. Jeu d’essai

- Première interface:

Cette interface affiche à l'utilisateur un message indiquant la mission du programme et ses créateurs. Ainsi Le menu est affiché comme suit :



Pour accéder aux fonctionnalités proposées dans le menu, l’utilisateur doit taper un nombre compris entre 1 et 9 sinon un message d’erreur s’affiche.

- Affichage de l'état de la mémoire centrale au début de l'exécution de notre programme :

```
--------> Veuillez selectionner une option: 7

        +*****************************************************************+
        |                                                                 |
        |                          Memoire Centrale                       |
        |                                                                 |
        +************++************++************++************++************+
        |            ||            ||            ||            ||            |
        +************++************++************++************++************+
        |            ||            ||            ||            ||            |
        +************++************++************++************++************+
        |            ||            ||            ||            ||            |
        +************++************++************++************++************+
        |            ||            ||            ||            ||            |
        +************++************++************++************++************+
        |            ||            ||            ||            ||            |
        +************++************++************++************++************+
        |            ||            ||            ||            ||            |
        +************++************++************++************++************+
```

- Chargement d'un premier programme : prog1 de taille 700 registres

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog1
--------> veuillez saisir la taille de ce programme: 700


--------------le programme a ete bien enregistre
```

- Chargement d'un deuxième programme : prog2 de taille 300 registres :

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog2
--------> veuillez saisir la taille de ce programme: 300


--------------le programme a ete bien enregistre
```

- Chargement d'un troisième programme : prog3 de taille 900 registres :

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog3
--------> veuillez saisir la taille de ce programme: 900


--------------le programme a ete bien enregistre
```

- Chargement d'un quatrième programme : prog4 de taille 500 registres :

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog4
--------> veuillez saisir la taille de ce programme: 500


--------------le programme a ete bien enregistre
```

- Affichage de la liste des programmes dans la mémoire :

```
--------> Veuillez selectionner une option: 3


----------La lise des programmes :

-----------------# prog1
-----------------# prog2
-----------------# prog3
-----------------# prog4
```

- Affichage de la liste des pages qu'occupe chaque programme dans la mémoire :

```
--------> Veuillez selectionner une option: 5


----------La Liste des programmes :

----------------# prog1  : 1  2  3  4  5  6  7

----------------# prog2  : 8  9  10

----------------# prog3  : 11  12  13  14  15  16  17  18  19

----------------# prog4  : 20  21  22  23  24
```

- Affichage de l'état de la mémoire centrale après l'ajout des 4 programmes :

```
--------> Veuillez selectionner une option: 7


       +*************************************************************************+
       |                                                                         |
       |                          Memoire Centrale                               |
       |                                                                         |
       +************+++************+++************+++************+++************+
       |   prog1    ||   prog1    ||   prog1    ||   prog1    ||   prog1    |
       +************+++************+++************+++************+++************+
       |   prog1    ||   prog1    ||   prog2    ||   prog2    ||   prog2    |
       +************+++************+++************+++************+++************+
       |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog3    |
       +************+++************+++************+++************+++************+
       |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog4    |
       +************+++************+++************+++************+++************+
       |   prog4    ||   prog4    ||   prog4    ||   prog4    ||            |
       +************+++************+++************+++************+++************+
```

- Affichage de la liste des pages libres en mémoire :

```
--------> Veuillez selectionner une option: 2


----------Les pages vides dans la memoire :

 25
```

- Chargement d'un cinquième programme : prog5 de taille 300 registres :
  - Impossible car il reste qu'une seule page libre

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog5
--------> veuillez saisir la taille de ce programme: 300

----------!!!taille insuportable!!!
```

- Terminer l'exécution du deuxième programme :

```
--------> Veuillez selectionner une option: 6

--------> Veuillez saisir le nom du programme a supprimer: prog2

--------le programme a ete bien supprime-------
```

- Affichage de l'état de la mémoire centrale après suppression du deuxième programme :

```
--------> Veuillez selectionner une option: 7


       +*************************************************************************+
       |                                                                         |
       |                          Memoire Centrale                               |
       |                                                                         |
       +*************++*************++*************++*************++*************+
       |   prog1     ||   prog1     ||   prog1     ||   prog1     ||   prog1     |
       +*************++*************++*************++*************++*************+
       |   prog1     ||   prog1     ||             ||             ||             |
       +*************++*************++*************++*************++*************+
       |   prog3     ||   prog3     ||   prog3     ||   prog3     ||   prog3     |
       +*************++*************++*************++*************++*************+
       |   prog3     ||   prog3     ||   prog3     ||   prog3     ||   prog4     |
       +*************++*************++*************++*************++*************+
       |   prog4     ||   prog4     ||   prog4     ||   prog4     ||             |
       +*************++*************++*************++*************++*************+
```

- Affichage de la liste des programmes dans la mémoire :

```
--------> Veuillez selectionner une option: 3


----------La lise des programmes :

-----------------# prog1
-----------------# prog3
-----------------# prog4
```

- Affichage de la liste des pages qu'occupe chaque programme dans la mémoire :

```
--------> Veuillez selectionner une option: 5


----------La Liste des programmes :

-----------------# prog1  : 1  2  3  4  5  6  7

-----------------# prog3  : 11  12  13  14  15  16  17  18  19

-----------------# prog4  : 20  21  22  23  24
```

- Affichage de la liste des pages libres en mémoire :

```
--------> Veuillez selectionner une option: 2


----------Les pages vides dans la memoire :

 8   9   10   25
```

- Terminer l'exécution du quatrième programme :

```
--------> Veuillez selectionner une option: 6

--------> Veuillez saisir le nom du programme a supprimer: prog4

--------le programme a ete bien supprime-------
```

- Affichage de l'état de la mémoire centrale après suppression du quatrième programme :

```
--------> Veuillez selectionner une option: 7

        +*************************************************************************+
        |                                                                         |
        |                          Memoire Centrale                               |
        |                                                                         |
        +************+++************+++************+++************+++************+
        |   prog1    ||   prog1    ||   prog1    ||   prog1    ||   prog1    |
        +************+++************+++************+++************+++************+
        |   prog1    ||   prog1    ||             ||             ||             |
        +************+++************+++************+++************+++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog3    |
        +************+++************+++************+++************+++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||             |
        +************+++************+++************+++************+++************+
        |            ||             ||             ||             ||             |
        +************+++************+++************+++************+++************+
```

- Affichage de la liste des programmes dans la mémoire :

```
--------> Veuillez selectionner une option: 3


----------La lise des programmes :

-----------------# prog1
-----------------# prog3
```

- Affichage de la liste des pages qu'occupe chaque programme dans la mémoire :

```
--------> Veuillez selectionner une option: 5

----------La Liste des programmes :
-----------------# prog1  : 1  2  3  4  5  6  7
-----------------# prog3  : 11  12  13  14  15  16  17  18  19
```

- Affichage de la liste des pages libres en mémoire :

```
--------> Veuillez selectionner une option: 2


----------Les pages vides dans la memoire :

 8   9   10   20   21   22   23   24   25
```

- Chargement d'un sixième programme : prog6 de taille 555 registres :

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog6
--------> veuillez saisir la taille de ce programme: 555


--------------le programme a ete bien enregistre
```

- Affichage de l'état de la mémoire centrale après l'ajout du sixième programme :

```
--------> Veuillez selectionner une option: 7

        +*************************************************************************+
        |                                                                         |
        |                          Memoire Centrale                               |
        |                                                                         |
        +************+++************+++************+++************+++************+
        |   prog1    ||   prog1    ||   prog1    ||   prog1    ||   prog1    |
        +************+++************+++************+++************+++************+
        |   prog1    ||   prog1    ||   prog6    ||   prog6    ||   prog6    |
        +************+++************+++************+++************+++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog3    |
        +************+++************+++************+++************+++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog6    |
        +************+++************+++************+++************+++************+
        |   prog6    ||   prog6    ||            ||            ||            |
        +************+++************+++************+++************+++************+
```

- Affichage de la liste des programmes dans la mémoire :

```
--------> Veuillez selectionner une option: 3


----------La lise des programmes :

-----------------# prog1
-----------------# prog3
-----------------# prog6
```

- Affichage de la liste des pages qu'occupe chaque programme dans la mémoire :

```
--------> Veuillez selectionner une option: 5


----------La Liste des programmes :

-----------------# prog1  : 1  2  3  4  5  6  7

-----------------# prog3  : 11  12  13  14  15  16  17  18  19

-----------------# prog6  : 8  9  10  20  21  22
```

- Affichage de la liste des pages libres en mémoire :

```
--------> Veuillez selectionner une option: 2


----------Les pages vides dans la memoire :

 23   24   25
```

- Terminer l'exécution du premier programme :

```
--------> Veuillez selectionner une option: 6

--------> Veuillez saisir le nom du programme a supprimer: prog1

--------le programme a ete bien supprime-------
```

- Affichage de l'état de la mémoire centrale après suppression du premier programme :

```
--------> Veuillez selectionner une option: 7

+**************************************************************************+
|                                                                          |
|                          Memoire Centrale                                |
|                                                                          |
+************+++************++************++************++************+
|            ||             ||            ||            ||            |
+************++************++************++************++************+
|            ||             ||   prog6     ||   prog6     ||   prog6     |
+************++************++************++************++************+
|   prog3    ||   prog3     ||   prog3     ||   prog3     ||   prog3     |
+************++************++************++************++************+
|   prog3    ||   prog3     ||   prog3     ||   prog3     ||   prog6     |
+************++************++************++************++************+
|   prog6    ||   prog6     ||            ||            ||            |
+************++************++************++************++************+
```

- Chargement d'un septième programme : prog7 de taille 1500 registres

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog7
--------> veuillez saisir la taille de ce programme: 1500

----------!!!taille insuportable!!!
```

- Chargement d'un septième programme : prog7 de taille 1000 registres

```
--------> Veuillez selectionner une option: 1

--------> Veuillez saisir le nom du programme a charger: prog7
--------> veuillez saisir la taille de ce programme: 1000


--------------le programme a ete bien enregistre
```

- Affichage de l'état de la mémoire centrale après l'ajout du septième programme :

```
--------> Veuillez selectionner une option: 7

        +*************************************************************************+
        |                                                                         |
        |                          Memoire Centrale                               |
        |                                                                         |
        +************+************+************+************+************+
        |   prog7    ||   prog7    ||   prog7    ||   prog7    ||   prog7    |
        +************++************++************++************++************+
        |   prog7    ||   prog7    ||   prog6    ||   prog6    ||   prog6    |
        +************++************++************++************++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog3    |
        +************++************++************++************++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||   prog6    |
        +************++************++************++************++************+
        |   prog6    ||   prog6    ||   prog7    ||   prog7    ||   prog7    |
        +************++************++************++************++************+
```

- Affichage de la liste des programmes dans la mémoire :

```
--------> Veuillez selectionner une option: 3


----------La lise des programmes :

-----------------# prog3
-----------------# prog6
-----------------# prog7
```

- Affichage de la liste des pages qu'occupe chaque programme dans la mémoire :

```
--------> Veuillez selectionner une option: 5

----------La Liste des programmes :
-----------------# prog3  : 11  12  13  14  15  16  17  18  19
-----------------# prog6  : 8  9  10  20  21  22
-----------------# prog7  : 1  2  3  4  5  6  7  23  24  25
```

- Affichage de la liste des pages libres en mémoire :

```
--------> Veuillez selectionner une option: 2

----------Aucune page vide n'existe dans la memoire !!!!
```

- Chercher un programme à partir de son nom :

  - Si le programme existe dans la mémoire, la liste des pages qu'il occupe s'affichera.

```
--------> Veuillez selectionner une option: 4
--------> Veuillez saisir le nom du programme a afficher: prog7

----------Les pages du programme prog7 :
           1  2  3  4  5  6  7  23  24  25
```

  - Sinon si le programme n'existe pas , un message d'erreur s'affichera

```
--------> Veuillez selectionner une option: 4
--------> Veuillez saisir le nom du programme a afficher: prog9
----------Ce programme n'existe pas !!!!
```

- Vider toute la mémoire centrale :

- Si la mémoire contient des programmes, c'est à dire elle n'est pas vide.

```
--------> Veuillez selectionner une option: 8

----------La mémoire a bien été vidé
```

- Si la mémoire est vide

```
--------> Veuillez selectionner une option: 8


----------La mémoire est dèja vide !!!!
```

- Affichage de l'état de la mémoire centrale après l'avoir vidée :

```
--------> Veuillez selectionner une option: 7

       +*************************************************************************+
       |                                                                         |
       |                          Memoire Centrale                               |
       |                                                                         |
       +*************++*************++*************++*************++*************+
       |             ||             ||             ||             ||             |
       +*************++*************++*************++*************++*************+
       |             ||             ||             ||             ||             |
       +*************++*************++*************++*************++*************+
       |             ||             ||             ||             ||             |
       +*************++*************++*************++*************++*************+
       |             ||             ||             ||             ||             |
       +*************++*************++*************++*************++*************+
       |             ||             ||             ||             ||             |
       +*************++*************++*************++*************++*************+
       |             ||             ||             ||             ||             |
       +*************++*************++*************++*************++*************+
```

- Affichage de la liste des programmes dans la mémoire après l'avoir vidée

```
--------> Veuillez selectionner une option: 3


----------la memoire est vide !!!
```

- Affichage de la liste des pages qu'occupe chaque programme dans la mémoire après l'avoir vidée :

```
--------> Veuillez selectionner une option: 5


--------!!!la memoire est vide!!!-------
```

- Affichage de la liste des pages libres après avoir vider la mémoire :

```
--------> Veuillez selectionner une option: 2

----------Les pages vides dans la memoire :
 1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25
```

- Le cas où l'utilisateur choisit une option que n'existe pas :

```
--------> Veuillez selectionner une option: 12

 !!!!!!!! veuillez saisir un numero depuis le menu !!!!!!!
```

- Quitter le programme :

```
--------> Veuillez selectionner une option: 9

 !!!!!!!! veuillez saisir un numero depuis le menu !!!!!!!


 *~~~~~~~~~~ Merci pour avoir utilise ce programme (^_^) a tres bientot ~~~~~~~~~~*


Process returned 0 (0x0)   execution time : 5375.335 s
Press any key to continue.
```

7. Partie supplémentaire:

a) Gestion de la mémoire par pagination avec l'ajout d'une mémoire virtuelle

La mémoire virtuelle est une fonctionnalité d'un système d'exploitation (OS, Operating System) qui permet à un ordinateur de compenser le manque de mémoire physique en transférant temporairement des pages de données de la mémoire vive (RAM, Random Access Memory) vers un stockage sur disque.

C'est pour cela que nous avons eu l'idée de faire une simulation de la gestion de mémoire en ajoutant le principe de la mémoire virtuelle.

b) Description des fonctionnalités de cette partie :

- **Ajouter un programme:**

L'ajout ici concerne l'ajout d'un programme dans la mémoire virtuelle**.**

- **Affichage de la mémoire virtuelle :**

Permet d'afficher les noms des programmes aves le nombre de pages qu'il occupe.

- **Exécuter un programme:**

Avant d'exécuter un programme, il doit exister dans la mémoire virtuelle. La liste des programmes qui existent dans celle-ci s'affiche pour pouvoir choisir l'un d'eux afin de l'ajouter dans la mémoire centrale.

- **Terminer l'exécution d'un programme:**

Le programme sera supprimé de la mémoire centrale mais il reste toujours dans la mémoire virtuelle.

- **Supprimer un programme:**

Le programme sera supprimé définitivement de la mémoire centrale et de la mémoire virtuelle.

NB : Les autres fonctionnalités de la gestion de la mémoire vont rester les mêmes.

c) Les algorithmes ajoutés :

Déclaration des structures de données :

Une structure de donnée s'ajoute (progV) qui définit la liste des programmes dans la mémoire virtuelle.

```
8
9
10  /******liste des pages*****/
11  struct page{
12      int adresse;
13      struct page *page_svt;
14  };
15
16  struct progV{
17    char nom[15];
18    int taille;
19    struct progV *suivant;
20
21  };
22
23  /******liste programme******/
24  struct prog{
25      char nom[15];
26      int taille;
27      struct page *page_prog;
28      struct prog *prog_svt;
29  };
30
```

Ajouter un programme:

La création d'une liste de programmes.

```
525 struct progV *CreerProgV(int Taille, char Name[15]) {
526
527     struct progV *Resultat = malloc(sizeof( struct progV));
528
529     Resultat->taille = Taille;
530
531     strcpy(Resultat->nom, Name);
532
533     Resultat->suivant = NULL;
534
535     return Resultat;
536 }
```

L'ajout d'un programme à la liste.

```
541 struct progV *AjouterProgV(struct progV *Precedent,struct progV *ProgrammeAjouter) {
542     if(Precedent!=NULL){
543         Precedent->suivant = ProgrammeAjouter;
544     }
545     return ProgrammeAjouter;
546 }
547
548
549
550 void afficher_pages_prog(struct prog *p,char l[]){
551
552     while(p!=NULL){
553
554         if(strcmp(l,p->nom)==0){
555         afficher_page(p->page_prog);
556          }
557         p=p->prog_svt;
558     }
559 }
```

Les étapes d'ajout d'un programme dans la mémoire virtuelle.

```
241         else if(strcmp(choix,"9")==0){
242
243                 printf("--------> veuillez saisir le nom du prog :");
244                 scanf("%s",&n);
245                 printf("--------> veuillez saisir la taille de ce programme: ");
246                 scanf("%d", &taille);
247         if(taille>0){
248                 int x=TrouverProgV(dv,n);
249             if(x==0){
250                 struct progV *tmpsv;
251                 tmpsv=CreerProgV(taille,n);
252                 AjouterProgV(pv,tmpsv);
253                 if(dv==NULL){
254                   dv=tmpsv;
255                 }
256                 pv=tmpsv;
257             }
258             else{
259                 printf("Ce programme existe deja");
260             }
261         }
262         else{
263             printf("taille inacceptable");
264         }
265         }
```

Supprimer un programme :

```
418 void SupprimerProgV(struct progV *head,char *nom){
419     struct progV *tmp = head;
420     struct progV *tmp2 = NULL;
421
422     if((strcmp(nom, head->nom) == 0)){
423         tmp = head;
424         head = head->suivant;
425         free(tmp);
426     }
427     else {
428         while ((strcmp(tmp->nom, nom)!=0)) {
429
430             tmp2=tmp;
431             tmp = tmp->suivant;
432         }
433         if((strcmp(tmp->nom, nom)==0)) {
434             tmp2->suivant = tmp->suivant;
435             free(tmp);
436         }
437     }
438 }
```

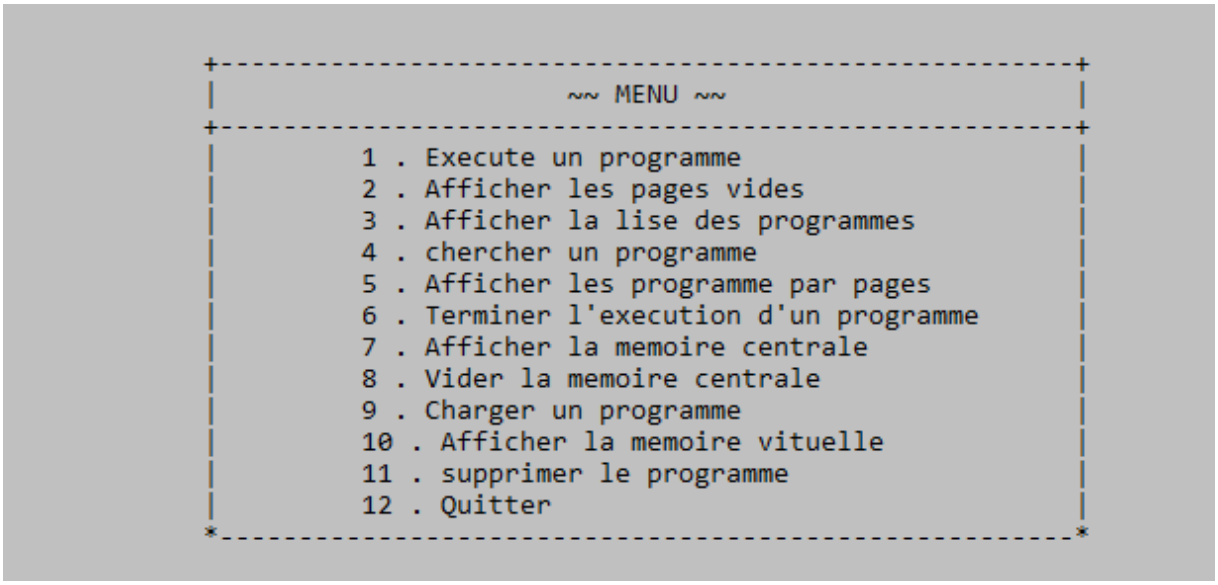
Si le programme existe dans la mémoire centrale il sera supprimé automatiquement d'abord de la mémoire centrale ensuite il sera supprimé de la mémoire virtuelle.

```
287         else if(strcmp(choix,"11")==0){
288             printf("--------> Veuillez saisir le nom du programme a supprimer: ");
289             scanf("%s", &n);
290             int v=charcherV(dv,n);
291             if(v==0){
292                 printf("--------ce programme n'existe pas");
293             }
294             else{
295                    int x=charcher(pgm,n);
296                  if(x==0){
297                     SupprimerProgV(dv,n);
298                  }
299                  else{
300                    pg=get_listpage(pgm,n);
301                    pgm=supprimer_prog(pgm, n);
302                    p=fusionner(p->page_svt,pg);
303                    SupprimerProgV(dv,n);
304                    printf("--------le programme a ete bien supprime-------\n");
305                  }
306
307             }
308
309         }
```

d) Jeu d'essai :

Première interface :

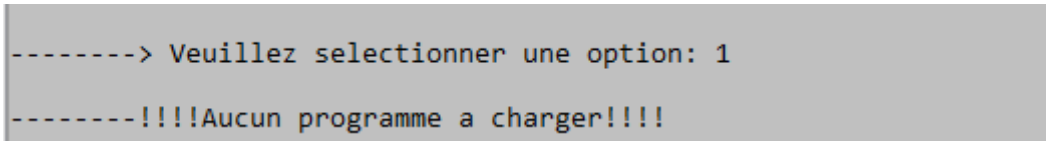
Cette interface affiche le menu.



Pour accéder aux fonctionnalités proposées dans le menu, l'utilisateur doit taper un nombre compris entre 1 et 12 sinon un message d'erreur s'affiche.

Exécuter un programme :

Un message d'erreur s'affiche car la mémoire virtuelle est vide.



- **Chargement de programmes :**

  Grace à l'option 9 on ajoute 4 programmes à la mémoire virtuelle.

- **Affichage de la mémoire virtuelle après l'ajout des 4 programmes :**

```
--------> Veuillez selectionner une option: 10

                +**********************************+
                |       La Memoire Virtuelle       |
                +**********************************+
                        prog1 : 3 pages
                +**********************************+
                        prog2 : 5 pages
                +**********************************+
                        prog3 : 7 pages
                +**********************************+
                        prog4 : 2 pages
                +**********************************+
```

- **Exécution des programmes prog4 et prog3 :**

Pour exécuter un programme qui existe déjà dans la mémoire virtuelle on doit le choisir depuis la liste des programmes se trouvant dans la mémoire virtuelle. Ensuite le programme sera exécuter dans la mémoire centrale.

```
--------> Veuillez selectionner une option: 1

--------------1 :prog1
--------------2 :prog2
--------------3 :prog3
--------------4 :prog4
-------->Entrer votre choix :4

--------------le programme a ete bien enregistre
```

```
--------> Veuillez selectionner une option: 1

--------------1 :prog1
--------------2 :prog2
--------------3 :prog3
--------------4 :prog4
-------->Entrer votre choix :3

--------------le programme a ete bien enregistre
```

- **Etat de la mémoire centrale après l'exécution du programme prog4 :**

```
--------> Veuillez selectionner une option: 7

      +*************************************************************************+
      |                                                                         |
      |                          Memoire Centrale                               |
      |                                                                         |
      +************+++************++************++************++************+
      |   prog4    ||   prog4    ||   prog3    ||   prog3    ||   prog3    |
      +************++************++************++************++************+
      |   prog3    ||   prog3    ||   prog3    ||   prog3    ||            |
      +************++************++************++************++************+
      |            ||            ||            ||            ||            |
      +************++************++************++************++************+
      |            ||            ||            ||            ||            |
      +************++************++************++************++************+
      |            ||            ||            ||            ||            |
      +************++************++************++************++************+
      |            ||            ||            ||            ||            |
      +************++************++************++************++************+
```

- **Terminer l'exécution du programme prog4 :**

```
--------> Veuillez selectionner une option: 6

--------> Veuillez saisir le nom du programme pour terminer l'execution: prog4
--------le programme a ete bien supprime-------
```

- **Affichage de l'état de la mémoire centrale après avoir terminé l'exécution du programme prog4 :**

On remarque que le programme prog4 est supprimé de la mémoire centrale .

```
--------> Veuillez selectionner une option: 7

        +*****************************************************************+
        |                                                                 |
        |                        Memoire Centrale                         |
        |                                                                 |
        +************++************++************++************++************+
        |           ||           ||   prog3    ||   prog3    ||   prog3    |
        +************++************++************++************++************+
        |   prog3    ||   prog3    ||   prog3    ||   prog3    ||           |
        +************++************++************++************++************+
        |           ||           ||           ||           ||           |
        +************++************++************++************++************+
        |           ||           ||           ||           ||           |
        +************++************++************++************++************+
        |           ||           ||           ||           ||           |
        +************++************++************++************++************+
        |           ||           ||           ||           ||           |
        +************++************++************++************++************+
```

- **Affichage de l'état de la mémoire virtuelle après avoir terminé l'exécution du programme prog4 :**

Même si le programme a été supprimé de la mémoire centrale, il reste toujours dans la mémoire virtuelle.

```
--------> Veuillez selectionner une option: 10

                +************************************+
                |       La Memoire Virtuelle         |
                +************************************+
                        prog1 : 3 pages
                +************************************+
                        prog2 : 5 pages
                +************************************+
                        prog3 : 7 pages
                +************************************+
                        prog4 : 2 pages
                +************************************+
```

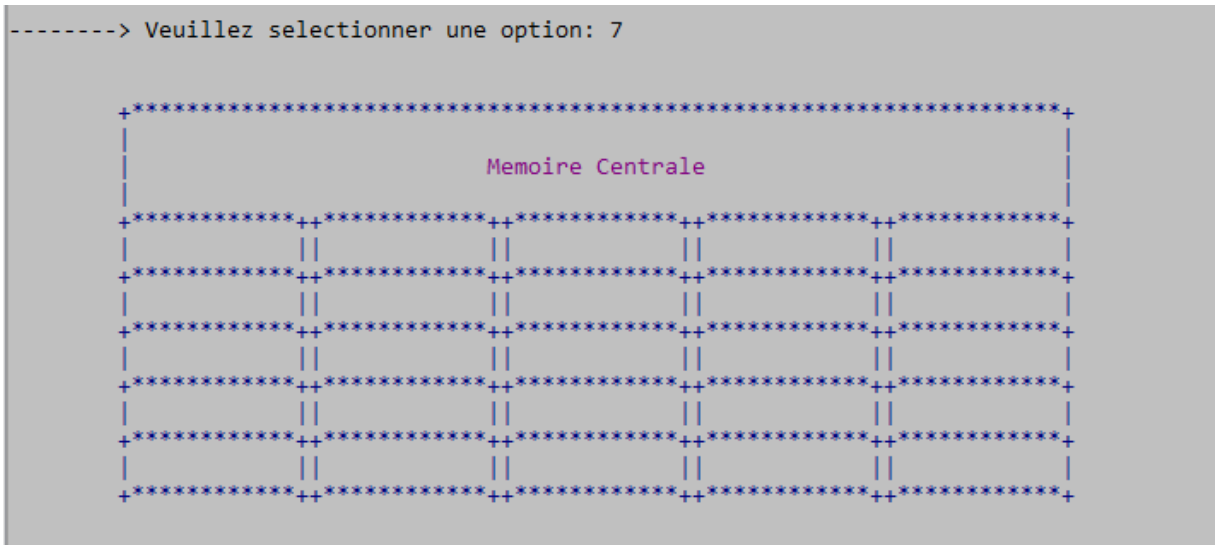
- **Supprimer le programme prog3 définitivement :**

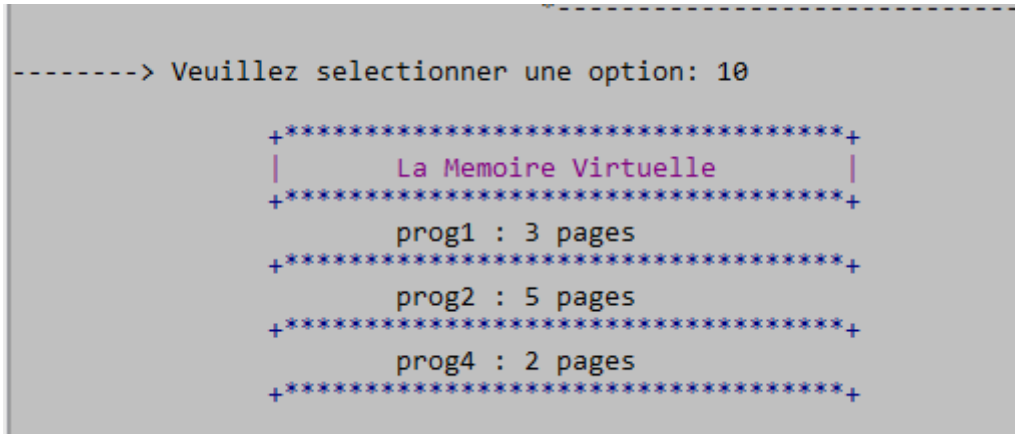
```
--------> Veuillez selectionner une option: 11

--------> Veuillez saisir le nom du programme a supprimer: prog3
--------le programme a ete bien supprime-------
```

- **Affichage de l'état de la mémoire centrale et la mémoire virtuelle après la suppression de programme prog3 :**

On remarque que le programme prog3 a été supprimé de la mémoire centrale et de la mémoire virtuelle





Conclusion

Ce travail a été une opportunité de mettre en exergue nos connaissances et nos compétences en ce qui concerne la gestion de mémoire, en étant l'une des bases de l'architecture des systèmes d'exploitation, et de les exploiter également à travers un projet concret en langage C. Ceci en utilisant la technique de pagination qui consiste sur la distribution de l'espace d'adressage entier sur des blocs de longueur fixe qui sont utilisés en tant que quantification mémoire.