Série 01

Exo 2

Partie 01

```
#include<stdio.h>

#include<sys/wait.h>

#include<sys/types.h>

#include<stdlib.h>

#include<unistd.h>

main (){

execlp("/ bin /  ls   ","ls",  NULL); }
```

Partie 02

```
#include<stdio.h>
#include<sys/wait.h>
#include<sys/types.h>
#include<stdlib.h>
#include<unistd.h>
main (){
int i;
printf(" 0- editer un fichier\n");
printf(" 1-  aficher un fichier\n");
printf(" 2- supprimer un fichier\n");
printf("  choix ? \n");
scanf("%d",&i);
switch(i){
 case(0):{execlp("/etc/alternatives/vi","vi","/home/an12/Desktop/essai1.c",NULL); break;}
case(1):{execlp("/bin/cat","cat","/home/an12/Desktop/exo2.c",NULL);break;}
 case(2):{execlp("/bin/rm","rm","/home/an12/Desktop/essai.c",NULL);break;}
default: printf("erreur \n");
}
}
```

Partie 03

```
    #include<stdio.h>
    #include<sys/wait.h>
    #include<sys/types.h>
    #include<stdlib.h>
    #include<unistd.h>
    main(){
    int i,j,k,s;
    printf(" premier nombre\n");
```

```
	scanf("%d",&i);
	printf(" deuxieme nombre\n");
	scanf("%d",&j);
	printf(" troisieme nombre\n");
	scanf("%d",&k);
	s=i+j+k;
	printf("la somme des trois nombres est %d \n",s);}
```

Programme appelant :

```
#include<stdio.h>
#include<sys/wait.h>
#include<sys/types.h>
#include<stdlib.h>
#include<unistd.h>

 main (){
char *argv[]={"somme","nbr_un","nbr_deux","nbr_trois",NULL};
execv("/home/an12/Desktop/somme.exe",argv);
}
```

Série 02

Exo 2

```
#include <stdio.h>
 #include <unistd.h>
 #include <stdlib.h>
#include <pthread.h>

//volatile
char theChar = '\0';
//volatile
char afficher = 0;

void* lire (void* name)
{ int cpt=0;
     while (theChar != 'F'){
        while (afficher == 1) {cpt ++;} /* attendre mon tour */
       scanf("%c",&theChar);
        afficher = 1; /* donner le tour */
    }
    return NULL;
}

void* affichage (void* name)
{
    int cpt = 0;
    while (theChar != 'F'){
                   while (afficher == 0) {} /* attendre */
```

```
            //if(afficher==1){
            printf("car = %c\n", theChar);
        afficher = 0; /* donner le tour */
            //printf("afficher= %d\n",afficher);
    //usleep(100);
            }// }
    return NULL;
}

int main (void)
{
    pthread_t filsA, filsB;

    if (pthread_create(&filsA, NULL, affichage, NULL)) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
    if (pthread_create(&filsB, NULL, lire, NULL)) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }

    if (pthread_join(filsA, NULL))
        perror("pthread_join");

    if (pthread_join(filsB, NULL))
        perror("pthread_join");
    printf("Fin du pere\n") ;

    return (EXIT_SUCCESS);

}
```

Exo 03

```
#include <stdio.h>

 #include <unistd.h>

 #include <stdlib.h>

#include <pthread.h>


int A[10][10],B[10][10],C[10][10];


//thread HD

void* addHD(void *k){

int i,j;

for (i=0; i<5; i++){

            for (j=5; j<10; j++){

                        C[i][j]=B[i][j]+A[i][j];

            }

}

}

// thread HG

void* addHG(void *k){

int i,j;

for (i=0; i<5; i++){

            for (j=0; j<5; j++){

                        C[i][j]=B[i][j]+A[i][j];
```

```
		}
	}
	}
	// thread BD
	void* addBD(void *k){
	int i,j;
	for (i=5; i<10; i++){
		for (j=5; j<10; j++){
			C[i][j]=B[i][j]+A[i][j];
		}
	}
	}
	//thread BG
	void* addBG(void *k){
	int i,j;
	for (i=5; i<10; i++){
		for (j=0; j<5; j++){
			C[i][j]=B[i][j]+A[i][j];
		}
	}


	}





	int main(){
```

```
int i,j;

pthread_t HG,HD,BG,BD;


//initialiser A

for (i=0; i<10; i++){

        for (j=0; j<10; j++){

                A[i][j] = 1;

        }

        }


//initialiser B

for (i=0; i<10; i++){

        for (j=0; j<10; j++){

                B[i][j] = i*10+j;

        }

}


// afficher la matrice B

for (i=0; i<10; i++){

        for (j=0; j<10; j++){

                printf("%d\t",B[i][j]);

        }

printf("\n");

}

// creer thread haut droite

if (pthread_create(&HD, NULL, addHD, NULL)) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
if (pthread_create(&HG, NULL, addHG, NULL)) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
if (pthread_create(&BD, NULL, addBD, NULL)) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
if (pthread_create(&BG, NULL, addBG, NULL)) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
//usleep(50);

//attendre fin
 if (pthread_join(HD, NULL))
        perror("pthread_join");
 if (pthread_join(HG, NULL))
        perror("pthread_join");
 if (pthread_join(BD, NULL))
        perror("pthread_join");
 if (pthread_join(BG, NULL))
        perror("pthread_join");
```

```
// aficher la atrice C
for (i=0; i<10; i++){
        for (j=0; j<10; j++){
                printf("%d\t",C[i][j]);
        }
printf("\n");
}

return 0;
}
```

Exo 04

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <math.h>
int Val1,Val2,Compt1=0,Compt2=0;
int tours=0;
pthread_mutex_t mutex;

void *J1(void *t){
Val1=rand()%3+1;
printf("%d\n",Val1);
}


void *J2(void *t){
```

```
Val2=rand()%3+1;

printf("%d\n",Val2);

}




void *Compter(void *t){

pthread_mutex_lock(&mutex);

                    if ((Val1==1) && (Val2==2))  Compt2++;

                    if ((Val1==2) && (Val2==1))  Compt2++;

                    if ((Val1==3) && (Val2==1))  Compt1++;


                    if ((Val1==1) && (Val2==3))  Compt1++;

                    if ((Val1==2) && (Val2==3))  Compt2++;

                    if ((Val1==3) && (Val2==2))  Compt1++;


pthread_mutex_unlock(&mutex);

}


int main(){

//Demande de tours

printf("Donner le nombre de tours");

scanf("%d",&tours);


pthread_mutex_init(&mutex,NULL);

pthread_t j1, j2,comp;
```

```
while (tours>0){
  pthread_create(&j1, NULL, J1, NULL);
  pthread_create(&j2, NULL, J2, NULL);
  pthread_create(&comp, NULL, Compter, NULL);



    if (pthread_join(j1, NULL))
        perror("pthread_join");

    if (pthread_join(j2, NULL))
        perror("pthread_join");

    if (pthread_join(comp, NULL))
        perror("pthread_join");

tours--;
printf("%d VS %d \n",Compt1,Compt2);
}
return 0;
}
```

Exo 05

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
```

```
int t[10];
sem_t vide,pleine;
int j=0;
int i=0;

void produire ()
{

t[j] = rand()%10;
printf ("valeur produite est : %d \n",t[j]);
j++;
if (j==10) j=0;

}

void consommer ( )
{

printf ("valeur consommer est : %d \n",t[i]);
i++;
if (i==10) i=0;
}

void* consom(void *k){

for (int k=0; k<20; k++){
        sem_wait(&pleine);
        consommer();
        sem_post(&vide);
        usleep(10);

}

}

void* prod(void *k){

for (int l=0; l<20; l++){

        sem_wait(&vide);
        produire();
        sem_post(&pleine);

}
}


int main(){

sem_init(&vide, 0, 10);
sem_init(&pleine, 0, 0);
```

```
pthread_t producteur,consommateur;


pthread_create(&producteur, NULL, prod, NULL);

pthread_create(&consommateur, NULL, consom, NULL);


 if (pthread_join(producteur, NULL))

        perror("pthread_join");

 if (pthread_join(consommateur, NULL))

        perror("pthread_join");

}
```

Série 03

Exo 01

```
#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <pthread.h>

Char[50]  chaine1,chaine2;

Char x;

Int tab[2];

void* consom(void *k){

close (tab[1]);

read(tab[0],chaine2,50) ;
```

```
printf("chaine recue= %s ",chaine2) ;

close (tab[0]);

}


void* prod(void *k)

{

int i=0;

close (tab[0]);

scanf("%x",&x) ;

while (x!= ")

{

Chaine1[i]=x;

scanf("%x",&x) ;

i++;

}

write(tab[1],chaine1,50) ;

close (tab[1]);

}

int main(){

pipe(tab) ;

pthread_t producteur,consommateur;


pthread_create(&producteur, NULL, prod, NULL);

pthread_create(&consommateur, NULL, consom, NULL);


 if (pthread_join(producteur, NULL))

        perror("pthread_join");
 if (pthread_join(consommateur, NULL))
        perror("pthread_join");
}
```

Exo 2

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
Char[50]  chaine1,chaine2,chaine3;
Int tab[2];
int main(){
pipe(tab) ;
int pid ,i,j;
pid=fork() ;
if (pid==0) // traitement du fils
{
close (tab[1]);
read(tab[0],chaine2,50) ;
for (i=0 ;i<strlen(chaine2) ;i++)
{
If ((chaine2[i] >'a')&& (chaine2[i] <'Z'))
{
chaine3[j]=chaine2[i];
j++;
}
}
printf("chaine affiché apres traitement= %s \n",chaine3) ;
close (tab[0]);


}
else if (pid==-1)  // echec du fork
{
printf("echec ! \n ") ;
}
else   // traitement du pere
{
close (tab[0]);
scanf("%s",&chaine1) ;
write(tab[1],chaine1,50) ;
close (tab[1]);
}


}
```