Ecole Supérieur en Informatique de Sidi Bel Abbès

Module: **Système d'exploitation 1** **1ère année Cycle Secondaire**
Semestre **: S1** Année universitaire: **2020-2021**

TD/TP **Edition des liens**

Exercice 1

Considérons les programmes ci-dessous :

prog.c			

00	55	push %ebp	extern void fact(void);
			extern void pgcd(void);
			extern int d;
			extern int f;
			int a=4;
			int b=36;
			int main() {
			fact();
			pgcd();
			b = f+d;
			a = f-d;
			return 0;
			}

01	89 e5	mov %esp,%ebp	
03	83 e4 f0	and $0xfffffff0,%esp	
06	e8 fc ff ff ff	call "**fact**"	
0b	e8 fc ff ff ff	call "**pgcd**"	
10	8b 15 00 00 00 00	mov "**f**",%edx	
16	a1 00 00 00 00	mov "**d**",%eax	
1b	8d 04 02	lea (%edx,%eax,1),%eax	
1e	a3 00 00 00 00	mov %eax, "**b**"	
23	8b 15 00 00 00 00	mov "**f**",%edx	
29	a1 00 00 00 00	mov "**d**",%eax	
2e	89 d1	mov %edx,%ecx	
30	29 c1	sub %eax,%ecx	
32	89 c8	mov %ecx,%eax	
34	a3 00 00 00 00	mov %eax,"**a**"	
39	b8 00 00 00 00	mov $0x0,%eax	
3e	89 ec	mov %ebp,%esp	
40	5d	pop %ebp	
41	c3	ret	

fact.c			

00	55	push %ebp	extern int a;
			int f;
			void fact(void)
			{
			int i;
			f=1;
			if (a<0) f=-1;
			else for(i=a; i>0; i--) f*=i;
			}

01	89 e5	mov %esp,%ebp	
03	83 ec 10	sub $0x10,%esp	
06	c7 05 00 00 00 00 01 00 00 00	movl $0x1,"**f**"	
10	a1 00 00 00 00	mov "**a**",%eax	
15	85 c0	test %eax,%eax	
17	79 0c	jns 25 <fact+0x25>	
19	c7 05 00 00 00 00 ff ff ff ff	movl $0xffffffff, "**f**"	
23	eb 22	jmp 47 <fact+0x47>	
25	a1 00 00 00 00	mov "**a"**,%eax	
2a	89 45 fc	mov %eax,-0x4(%ebp)	
2d	eb 12	jmp 41 <fact+0x41>	
2f	a1 00 00 00 00	mov "**f**",%eax	
34	0f af 45 fc	imul -0x4(%ebp),%eax	
38	a3 00 00 00 00	mov %eax,"**f**"	
3d	83 6d fc 01	subl $0x1,-0x4(%ebp)	
41	83 7d fc 00	cmpl $0x0,-0x4(%ebp)	
45	7f e8	jg 2f <fact+0x2f>	
47	c9	leave	
48	c3	ret	

pgcd.c			

00	55	push %ebp	extern int b;
			extern int f;
			int d;
			void pgcd(void)
			{
			int i,j;
			d=-1;
			if(b>0 && f>0){
			i=b;
			j=f;
			while(i!=j){
			if(i>j)
			i-=j;
			else
			j-=i;
			}
			d=i;
			}
			}

01	89 e5	mov %esp,%ebp	
03	83 ec 10	sub $0x10,%esp	
06	c7 05 00 00 00 00 ff ff ff ff	movl $0xffffffff, "**d**"	
10	a1 00 00 00 00	mov "**b"**,%eax	
15	85 c0	test %eax,%eax	
17	7e 41	jle 5a <pgcd+0x5a>	
19	a1 00 00 00 00	mov "**f"**,%eax	
1e	85 c0	test %eax,%eax	
20	7e 38	jle 5a <pgcd+0x5a>	
22	a1 00 00 00 00	mov "**b"**,%eax	
27	89 45 f8	mov %eax,-0x8(%ebp)	
2a	a1 00 00 00 00	mov "**f"**,%eax	
2f	89 45 fc	mov %eax,-0x4(%ebp)	
32	eb 16	jmp 4a <pgcd+0x4a>	
34	8b 45 f8	mov -0x8(%ebp),%eax	
37	3b 45 fc	cmp -0x4(%ebp),%eax	
3a	7e 08	jle 44 <pgcd+0x44>	
3c	8b 45 fc	mov -0x4(%ebp),%eax	
3f	29 45 f8	sub %eax,-0x8(%ebp)	
42	eb 06	jmp 4a <pgcd+0x4a>	
44	8b 45 f8	mov -0x8(%ebp),%eax	
47	29 45 fc	sub %eax,-0x4(%ebp)	
4a	8b 45 f8	mov -0x8(%ebp),%eax	
4d	3b 45 fc	cmp -0x4(%ebp),%eax	
50	75 e2	jne 34 <pgcd+0x34>	
52	8b 45 f8	mov -0x8(%ebp),%eax	
55	a3 00 00 00 00	mov %eax,"**d**"	
5a	c9	leave	
5b	c3	ret	

Et les modules objets « ***simplifiés*** » obtenus après compilation des programmes : prog.c , fact.c et pgcd.c :

prog.o :

Tables des symboles			
identificateur	**type**	**taille**	**déplacement**
a	.data	00000004	00000000
b	.data	00000004	00000000
main	.text	00000042	00000000
fact	REF	00000000	00000000
pgcd	REF	00000000	00000000
f	REF	00000000	00000000
d	REF	00000000	00000000

Table des Translations	
identificateur	**Adresses à translater**
fact	00000007
pgcd	0000000c
f	00000012, 00000025
d	00000017, 0000002a
b	0000001f
a	00000035

	Code			
00000000	55 89 e5 83	e4 f0 e8 fc	ff ff ff e8	fc ff ff ff
00000010	8b 15 00 00	00 00 a1 00	00 00 00 8d	04 02 a3 00
00000020	00 00 00 8b	15 00 00 00	00 a1 00 00	00 00 89 d1
00000030	29 c1 89 c8	a3 00 00 00	00 b8 00 00	00 00 89 ec
00000040	5d c3			

fact.o :

Tables des symboles			
identificateur	**type**	**taille**	**déplacement**
f	.BSS	00000004	00000000
fact	.text	00000049	00000000
a	REF	00000000	00000000

Table des Translations	
identificateur	**Adresses à translater**
f	00000008, 0000001b, 00000030, 00000039
a	00000011, 00000026

	Code			
00000000	55 89 e5 83	ec 10 c7 05	00 00 00 00	01 00 00 00
00000010	a1 00 00 00	00 85 c0 79	0c c7 05 00	00 00 00 ff
00000020	ff ff ff eb	22 a1 00 00	00 00 89 45	fc eb 12 a1
00000030	00 00 00 00	0f af 45 fc	a3 00 00 00	00 83 6d fc
00000040	01 83 7d fc	00 7f e8 c9	c3	

pgcd.o :

Tables des symboles			
identificateur	**type**	**taille**	**déplacement**
d	.BSS	00000004	00000000
pgcd	.text	0000005c	00000000
b	REF	00000000	00000000
f	REF	00000000	00000000

Table des Translations	
identificateur	**Adresses à translater**
d	00000008, 00000056
b	00000011, 00000023
f	0000001a, 0000002b

	Code			
00000000	55 89 e5 83	ec 10 c7 05	00 00 00 00	ff ff ff ff
00000010	a1 00 00 00	00 85 c0 7e	41 a1 00 00	00 00 85 c0
00000020	7e 38 a1 00	00 00 00 89	45 f8 a1 00	00 00 00 89
00000030	45 fc eb 16	8b 45 f8 3b	45 fc 7e 08	8b 45 fc 29
00000040	45 f8 eb 06	8b 45 f8 29	45 fc 8b 45	f8 3b 45 fc
00000050	75 e2 8b 45	f8 a3 00 00	00 00 c9 c3	

1) On suppose que l'adresse de début de la section code (**.text**) est **0x00000000**, l'adresse de début de la section des données initialisées(**.data**) est **0x00000400** et l'adresse de début de la section des données non initialisée(**.BSS**) est **0x00000500**.

Remarque : la taille réelle d'une fonction doit être un multiple de 4 octets : On complète les fonctions par des instructions nop (code 0x90).

Passe1

a. Créer une table des modules (ordre de lecture des modules : prog.o , fact.o et pgcd.o)

Nom du module objet	Taille réelle en octets de la section .text	Adresse par rapport au début

b. Créer la table globale des symboles

Tables des symboles			
Identificateur	Type(.text, .data, .BSS , UND)	Taille	Adresse

b. Créer la table globale des translations

Identificateur	Liste des **toutes les** adresses à translater

2) Faire l'édition de liens des modules objet . Le format « simplifié » du fichier exécutable est la suivante :

En-tête nom du fichier ; adresse début d'exécution (1ère instruction du programme principal) ; adresse de la section de code ; taille total de la section code ; adresse et taille de la section données initialisées, adresse et taille de la section données non initialisées(BSS)
Code (les instructions) : **.text**
Données initialisées : .data
Données non initialisées : .BSS

Passe2 Résoudre les références externes et faire les translations des adresses du fichier exécutable ci-dessous.

prog; 00000000; 00000000;. . . . . ; 00040000, ...; 00050000, ...;				
00000000	5589e583	e4f0e8- -	- - - - - - e8	- - - - - - - -
00000010	8b15. . . .	. . . . a1. .	. . . . . 8d	0402a3. .
00000020	. . . . . 8b	15.	. . a1. . . .	. . . . 89d1
00000030	29c189c8	a3.	. . b80000	000089ec
00000040	5dc39090	5589e583	ec10c705	. . . . .
00000050	01000000	a1.	. . 85c079	0cc705. .
00000060	. . . . . ff	ffffffeb	22a1. . . .	 8945
00000070	fceb12a1	. . . . .	0faf45fc	a3. . . . .
00000080	. . 836dfc	01837dfc	007fe8c9	c3909090
00000090	5589e583	ec10c705	. . . . .	ffffffff
000000a0	a1.	. . 85c07e	41a1. . . .	. . . . 85c0
000000b0	7e38a1. .	 89	45f8a1. .	 89
000000c0	45fceb16	8b45f83b	45fc7e08	8b45fc29
000000d0	45f8eb06	8b45f829	45fc8b45	f83b45fc
000000e0	75e28b45	f8a3. . . .	. . . . c9c3	
…				
00000400	00000004	00000024		/*.DATA
…				
00000500				/*.BSS

Exercice2 : TP

1) Créer un répertoire EDL et un répertoire BIB

2) Ecrire deux fonctions (dans le répertoire EDL) Calcul du factorielle d'un nombre entier >= 0 : long int fact(short int n) ; Calcul du PGCD de 2 nombres entiers > 0 : int pgcd(int a, int b) ;

3) Compiler et créer les modules objets correspondant à ces deux fonctions

4) Créer dans le répertoire **BIB** une bibliothèque dynamique **libfpg.so** contenant les 2 fonctions **fact** et **pgcd**.

5) Créer dans le répertoire **BIB** une bibliothèque statique **libfpg.a** contenant les 2 fonctions **fact** et **pgcd**.

6) Ecrire un programme **prog.c**, dans le répertoire **EDL** qui fait appel à **fact** et **pgcd**

a) Compiler **prog.c** et créer un module objet **prog.o** ,
b) Faire l'édition de liens de **prog.o** avec la bibliothèque dynamique **libfpg.so** et créer un fichier exécutable **progd**,
c) Faire l'édition de liens de **prog.o** avec la bibliothèque statique **libfpg.a** et créer un fichier exécutable **progs**,

7) Comparer les tailles des fichiers exécutables **progd** et **progs**.

8) Donner les séquences d'instructions de la table **.plt** qui permettent de faire l'édition de liens dynamique des fonctions **fact , pgcd** (ainsi que les autres fonctions utilisées dans prog.c telles printf,…) et le contenu des mots correspondants à ces fonctions dans la table **.got.plt** Utiliser la commande objdump (options –d et –s ; Pour plus de détails voir le man) . **Remarque** vous pouvez tester d'autres commandes : readelf, size, nm.