Assembleur et débogage

1 Registres du processeur x64

-	*nom*	*nb. bits*	*signification*	*rôle*
-	**rip**			
-	**rflags**			
-	**rsp**			
-	**rbp**			
14 registres généraux	**rax** ...**rdx**, **r8** ...**r15**		numérotés	
8 registres nombres flottants	**fpr0** ...**fpr7**			
16 registres étendus	**xmm0** ...**xmm15**			
autres	-	-	-	réservés au système d'exploitation

Remarque : les registres sont souvent subdivisés en 2 : **eax**, **edi**...

2 Assembleur

— le format est : **instruction destination, source**;
— en réalité, le code est sous forme binaire illisible par un humain.

instruction	*signification*	*rôle*
nop		
mov		
push, pop	-	
add, sub, mul , div	-	
inc		
cmp		
jmp, jle, jge...		
call	-	
lea		
ret		

```
#include <stdio.h>

int main(void) {
    for (int i = 0; i < 10; ++i) {
        printf("Bonjour %d !\n", i);
    }
    return 0;
}
```

```
0000000000000615 <main>:
 615:   55                      push   rbp
 616:   48 89 e5                mov    rbp,rsp
 619:   48 83 ec 10             sub    rsp,0x10
 61d:   c7 45 fc 00 00 00 00    mov    DWORD PTR [rbp-0x4],0x0
 624:   eb 1a                   jmp    640 <main+0x2b>
 626:   8b 45 fc                mov    eax,DWORD PTR [rbp-0x4]
 629:   89 c6                   mov    esi,eax
 62b:   48 8d 3d a2 00 00 00    lea    rdi,[rip+0xa2]
 632:   b8 00 00 00 00          mov    eax,0x0
 637:   e8 e4 fe ff ff          call   520 <printf@plt>
 63c:   83 45 fc 01             add    DWORD PTR [rbp-0x4],0x1
 640:   83 7d fc 09             cmp    DWORD PTR [rbp-0x4],0x9
 644:   7e e0                   jle    626 <main+0x11>
 646:   b8 00 00 00 00          mov    eax,0x0
 64b:   c9                      leave
 64c:   c3                      ret
 64d:   0f 1f 00                nop    DWORD PTR [rax]
```

3 Débogueur (gdb)

Compiler avec l'option [ ], puis lancer [ ] :

```
gcc -g bonjour.c -o bonjour.out
gdb ./bonjour.out
```

instruction	*raccourci*	*rôle*
`list`	`-`	
`disassemble f`	`disass f`	
`breakpoint f`	`b f`	
`breakpoint 18`	`b 18`	
`run <params>`	`r <params>`	
`continue`	`c`	
`next`	`n`	
`nexti`	`ni`	
`step`	`s`	
`stepi`	`si`	
`backtrace`	`bt`	
`info registers`	`i r`	
`info register rip`	`i r rip`	
`x 0x7fffffffdd36`	`-`	
`quit`	`q`	

Options d'examen de la mémoire

— nombre de blocs + taille du bloc + format
— exemple : **`x/12gx $rsp`**

taille	signification	traduction	nombre de bits
`b`			
`w`			
`g`			

format	signification
`x`	
`d`	
`f`	
`s`	
`i`	

4 Organisation de la mémoire