

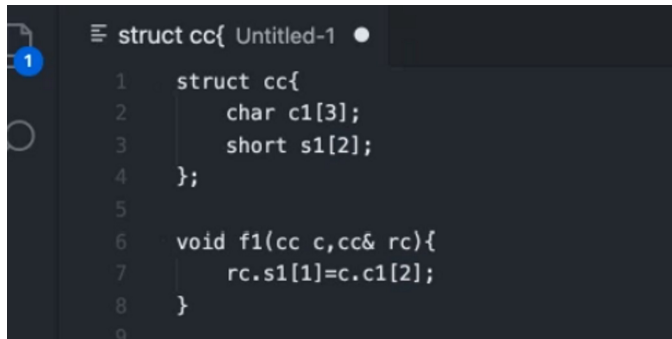
Domande orale 10/06/2021

giovedì 10 giugno 2021 09:26

Ad inizio orale il prof mostra al candidato la prova pratica svolta e se ha qualche dubbio a riguardo può chiedere chiarimenti, quindi è consigliato imparare la teoria dietro agli errori commessi durante la prova pratica e la loro soluzione

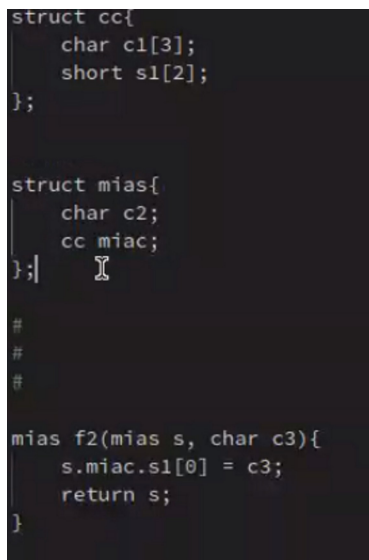
Corrispondenza = il prof ti detta una o più strutture e/o una classe, una o due funzioni e bisogna tradurle in assembler. Bisogna spiegarli cosa stiamo facendo mentre svolgiamo l'esercizio. L'esercizio viene fatto condividendo il proprio schermo e svolgendo l'esercizio su un editor a scelta.

1) corrispondenza



```
1 struct cc{
2     char c1[3];
3     short s1[2];
4 };
5
6 void f1(cc c, cc& rc){
7     rc.s1[1]=c.c1[2];
8 }
9
```

- 2) Inizializzazione descrittore di processo quando creiamo un processo
- 3) Possibile che arrivi un page fault sulla pila utente da parte di un processo di livello sistema
- 4) Corrispondenza



```
struct cc{
    char c1[3];
    short s1[2];
};

struct mias{
    char c2;
    cc miac;
};

#
#
#

mias f2(mias s, char c3){
    s.miac.s1[0] = c3;
    return s;
}
```

- 5) passaggio da livello utente a livello sistema e viceversa, cambio pila e controlli, motivare il cambio e i controlli da fare
- 6) Perché usiamo 12 bit di offset? Da dove deriva il numero 12? Lo abbiamo deciso no oppure ci è imposto?
- 7) Se avessimo voluto un offset più grande bisogna riadattarci a qualcosa?
- 8) Se avessimo avuto pagine di 2mib nel compito, avremmo avuto altri vincoli rispetto a quelli delle pagine di 4kib quando allocchiamo frame ?
- 9) Corrispondenza

```

struct cc{
    short s[4];
};

int f1 (cc &c1) {
    return c1.s[0]++;
}

int f2 (cc c2) {
    f1(c2);
    return c2.s[0];
}

```

10) È possibile fare l'ottimizzazione data dalla red zone in f2?

11) Corrispondenza

```

class c{
    int a[2];
    long b[2];

    c(int i);
}

+7 +6 +5 +4 +3 +2 +1 0
+-----+
|                                     ||
+-----+

c::c(int i) {
    for(j = 0; j < 2; j++) {
        a[j] = b[j] = i;
    }
}

int c::f1(c& c1) {
    c c2(3);
    return b[0] + c2.a[1] + c1.a[0];
}

```

12) ingresso , uscita: controlli e vincoli sul buffer che l'utente passa alle primitive

13) i controlli vanno fatti anche in DMA? Che altri tipi di controlli bisogna fare?

14) se avessi ricreato la pila usando come bottom l'indirizzo che ha causato il page fault

15) corrispondenza

```

class c {
    short s[4];
    char c1;
    c& f1(short *s1);
};

c& c::f1(short *s1){
    for (i = 0; i < 4; i++){
        if (c1 < s1[i])
            s[i] = s1[i];
    }
    return *this;
}

```

16) TLB: informazione contenute e quando va invalidato

17) Corrispondenza

```
struct s {
    char a[6];
};

class c {
    int j;
    s s1;

    c f1( s s2){
        for(int i = 0; i < 6; i++){
            s1.a[i]=s2.a[i];
        }

        return *this;
    }
}
```

18) Quando bisogna invalidare tutto il tlb?

19) A cosa serve avere i bit in ogni entrata dei vari livelli delle tabelle di traduzione se alla fine conta solo il risultato complessivo?

Ad esempio se trovo il bit p a 0 mi posso fermare prima oppure dividere i livelli successivi in sola lettura resettando il bit R/W

20) corrispondenza

```
class c{
    int i;
    c *p;

    c* f1(int j);
}

c* c::f1(int j){
    c* p1 = this;
    for(;;){
        if(p1->i == j)
            return p1;
        p1 = p1->p;
    }
}
```

21) tlb: quando ci sono pagine di dimensioni diverse, il tlb visto può funzionare con le pagine grandi 4mib?

22) tlb: bit A, cosa serve e come funziona, come lo salvo nel tlb?

23) Corrispondenza

```
struct s {
    int i1;
    int i2;
};

struct s1 {
    char c;
    s s2;
};

s1 f1(s mias) {
    s1 tmp;
    tmp.s2 = mias;
    return tmp;
}
```

24) Struttura generale di un processo esterno (scrivere il codice della definizione nell'editor)

25) Invio dell'EOI senza utilizzo della WFI nel modulo IO, cosa bisogna fare?

26) Cosa succede se invio EOI prima di wfi() ?

27) se volessimo riprenderci la memoria che un processo non usa? Invece di aggiungerla quando serve

28) Corrispondenza

```
class c {
    long a[8];

    c f1();
};

c c::f1() {
    c temp;
    temp.a[0] = 1;
    temp.a[2] = 2;

    return temp;
}
```

29) cosa avrebbe comportato la RVO nell'esercizio ?

30) corrispondenza

```
class c{
    long a[8];

    c(const c& ){

    }

    c f1(c& c1){
        c1.a[3] += a[3];
        return c1;
    }
};
```

31) A cosa serve la cache e come funziona

32) Nelle cache associative in caso di miss con che criterio rimpiazzo?

33) Corrispondenza

Temp = i corrisponde a temp(i)

```
class c{
    char c;
    long l1;
    long l2;

    c (int i){
        c = l1 = l2 = i;
    }

    c f1(int i){
        c temp = i;
        return temp;
    }
};
```

34) problemi di interazione tra cache e bus mastering

35) Operazione di lettura in DMA, problemi e soluzione

36) Schema assembler generale di una primitiva(codice da scrivere sull'editor)

37) La iretq nello schema, su che pila agisce?

38) corrispondenza

```
class c{  
  
    int a[8];  
  
    c (int i){  
        for(int j=0; j < 8; j++){  
            a[j]=i;  
        }  
    }  
  
    c f1(){  
        c temp(3);  
        return c;  
    }  
};
```

39) Differenza tra questa versione e quella in cui f1() è

```
c f1(){  
    return temp(3);  
}
```

40) Problemi di interazioni tra cache e bus master e soluzione

41) Corrispondenza traduzione f2

```
class c{  
    long a[8];  
  
    c(long l) {}  
  
    c f1() {  
        return c(10);  
    }  
  
    void f2() {  
        c cl = f1();  
    }  
};
```

42) Corrispondenza

```

class c{
    long a[8];

    c(long l) {}

    c f1() {
        return c(10);
    }

    void f2() {
        c c1 = f1();
    }
}

```

- 43) Interruzioni: a cosa bisogna stare attenti quando si programma con le interruzioni?
- Le istruzioni che ci permettono di usare le interruzioni
 - Il compilatore non sa che esistono le interruzioni
 - In che modo bisogna dichiarare una variabile non inizializzata nel programma in modo che il compilatore non la ignori(dichiarazione volatile)

44) Corrispondenza

```

class c{
    long a[8];

    c f1(c& c1);
}

c c::f1(c& c1){
    c c2;
    for(int i=0;i<8;i++){
        c2.a[i] = a[i] - c1.a[i];
    }

    return c2;
}

```

- 45) Controllore APIC: descrizione e funzionamento
- Quando si accodano le richieste di interruzione su irr?
 - Come bisogna gestire le periferiche che condividono lo stesso Piedino dell'apic? (or wire)
 - come facciamo ad assicurarci che il software non perda le richieste di interruzioni nel caso precedente in cui più periferiche condividono lo stesso piedino? (usando come l'apic riconosce le richieste (sul fronte o sul livello), EOI)

46) Corrispondenza

```

class c{
    long a[8];

    c& f1(long *l){
        for(int i=0; i<8;i++){
            a[i]+=l[i];
        }
        return *this;
    }
}

```

- Che succede se non allineiamo a 16 rsp non sottraendo 32? avrei problemi? (no, avrebbe funzionato lo stesso, Ci sono delle istruzioni che però pretendono l'allineamento)

47) Che problemi ci risolve la paginazione?

- Che problema ho se ricarico un processo dopo che era stato swappato?

48) Corrispondenza

```

struct s {
    char c1, c2;
    int a[4];
}

s f1(s& s1) {
    s s2;
    s2.c1 = s1.c1;
    s2.a[0] = s1.c2;
    return s2;
}

```

49) Come mai creiamo la finestra nella nostra memoria? Che problemi mi risolve?

- Le tabelle di traduzione vengono allocate nella memoria fisica

50)problemi tra bus mastering e cache e risoluzione:

bus mastering interagisce con la memoria centrale, ma potrebbe essere non aggiornata perchè c'è la cache

```

class c{
    long a[8];

    c(long l){};

    c f1(){
        return c(10);
    }

    void f2(){
        c c1= f1();
    }
}

```

SOLUZIONE:

```

_ZN2c12f2Ev:
prologo
sub $8, %rsp
mov %rdi, -8(%rbp)
lea -72(%rbp), %rdi
call _ZN1c2f1Ev

```

```

_ZN1c2f1Ev:
prologo
sub $16, %rsp
mov %rdi, -8(%rbp)
mov $10, %rsi
call _Z1c1E1
mov -8(%rbp), %rax
leave
ret

```

stessa domanda di prima:

SE LA cache è writeback può essere che il dato non sia aggiornata
-> via software, prima dell'operazione di dma di fare una copia
dalla cache in ram -> via hardware ?mandando una richiesta alla
cache direttamente?

Ponte pci chiede al controllore cache una lettura e fa il merge
tra quello che c'è in cache e quello che vuole fare l'op. di dma

51) parte assembler di una primitiva:

```

.extern c_fun
a_fun:
call salvastato
call c_fun
call caricastato
iretq

```

la iretq su quale pila agisce? nella pila sistema del processo messo in
atto da c_fun.

come faccio a capire se la iretq ha tutte le informazioni che gli
necessitano?

la iretq deve obbligatoriamente passare a utente o può rimanere in mod.
sistema?

In quale pila viene salvato l'indirizzo di ritorno? nella pila sistema di
p1, quindi la carica_stato come fa a tornare in P1 se è in azione P2?
/* da qui in avanti non ho capito più nulla. il collega sbagliava e il
prof cambiava esempi per fargli arrivare alla soluzione, penso che la
domanda si riassume in cosa fa la carica_stato*/

52) in quale gate della idt abbiamo DPL sistema? nelle routine delle eccezioni o
int esterne


```
53) struct s{  
    short a[3];  
}
```

```
class c{  
    s s1;  
    s s2;
```

```
    int f1(s &s3){  
        if(s3.a[0] >1){return f1(s1);}  
        else return 0;  
    }  
};
```

54) interruzioni, quando si programma con le interruzioni a cosa bisogna stare attenti?