Tutorial 5.3

nested struct & pointer arguments

Lei Wang

lei.wang2@ucalgary.ca

nested struct

- a structure contains another structure
- Eg: C code

```c
struct struct1{
        int i_1;
        int j_1;
};

struct struct2{
        long int i_2;
        long int j_2;
};

struct nested_struct{
        int i_3;
        struct struct1 s_1;
        struct struct2 s_2;
};
```

assembly

```
//Equates for struct1
i1_s = 0
j1_s = 4
struct1_size = 8

//Equates for struct2
i2_s = 0
j2_s = 8
struct2_size = 16

//Equates for nested struct
i3_s = 0
struct1_s = 4
struct2_s = 12
nested_struct_size = 28
```


UNIVERSITY OF CALGARY

initialization

- Eg:

```
struct nested_struct init()
{
        struct nested_struct s;
        s.i_3 = 0
        s.struct1.i_1 = 0;
        s.struct1.j_1 = 0;
        s.struct2.i_2 = 0;
        s.struct2.j_2 = 0;
        return s;
}
```

```
                // function: init
                alloc = -(16 + nested_struct_size) & -16
                dealloc = -alloc
                n_s = 16
init:           stp     x29, x30, [sp, alloc]!
                mov     x29, sp
                // initialize the nested struct
                str     wzr, [x29, n_s + i3_s]
                str     wzr, [x29, n_s + struct1_s + i1_s]
                str     wzr, [x29, n_s + struct1_s + j1_s]
                str     xzr, [x29, n_s + struct2_s + i2_s]
                str     xzr, [x29, n_s + struct2_s + j2_s]
                //return initialized value by copying it
                  into memory at address in x8
                ldr     w9, [x29, n_s + i3_s]
                str     w9, [x8, i3_s]
                ldr     w9, [x29, n_s + struct1_s + i1_s]
                str     w9, [x8, struct1_s + i1_s]
                ldr     w9, [x29, n_s + struct1_s + j1_s]
                str     w9, [x8, struct2_s + j1_s]
                ldr     x9, [x29, n_s + struct2_s + i2_s]
                str     x9, [x8, struct2_s + i2_s]
                ldr     x9, [x29, n_s + struct2_s + j2_s]
                str     x9, [x8, struct2_s + j2_S]

                ldp     x29, x30, [sp], dealloc
                ret
```


UNIVERSITY OF CALGARY

example code

```
struct struct1{
        int i_1;
        int j_1;
};

struct struct2{
        long int i_2;
        long int j_2;
};

struct nested_struct{
        int i_3;
        struct struct1 s_1;
        struct struct2 s_2;
};

struct nested_struct init()
{
        struct nested_struct n;
        n.i_3 = 0;
        n.s_1.i_1 = 0;
        n.s_1.j_1 = 0;
        n.s_2.i_2 = 0;
        n.s_2.j_2 = 0;
        return n;
}
```

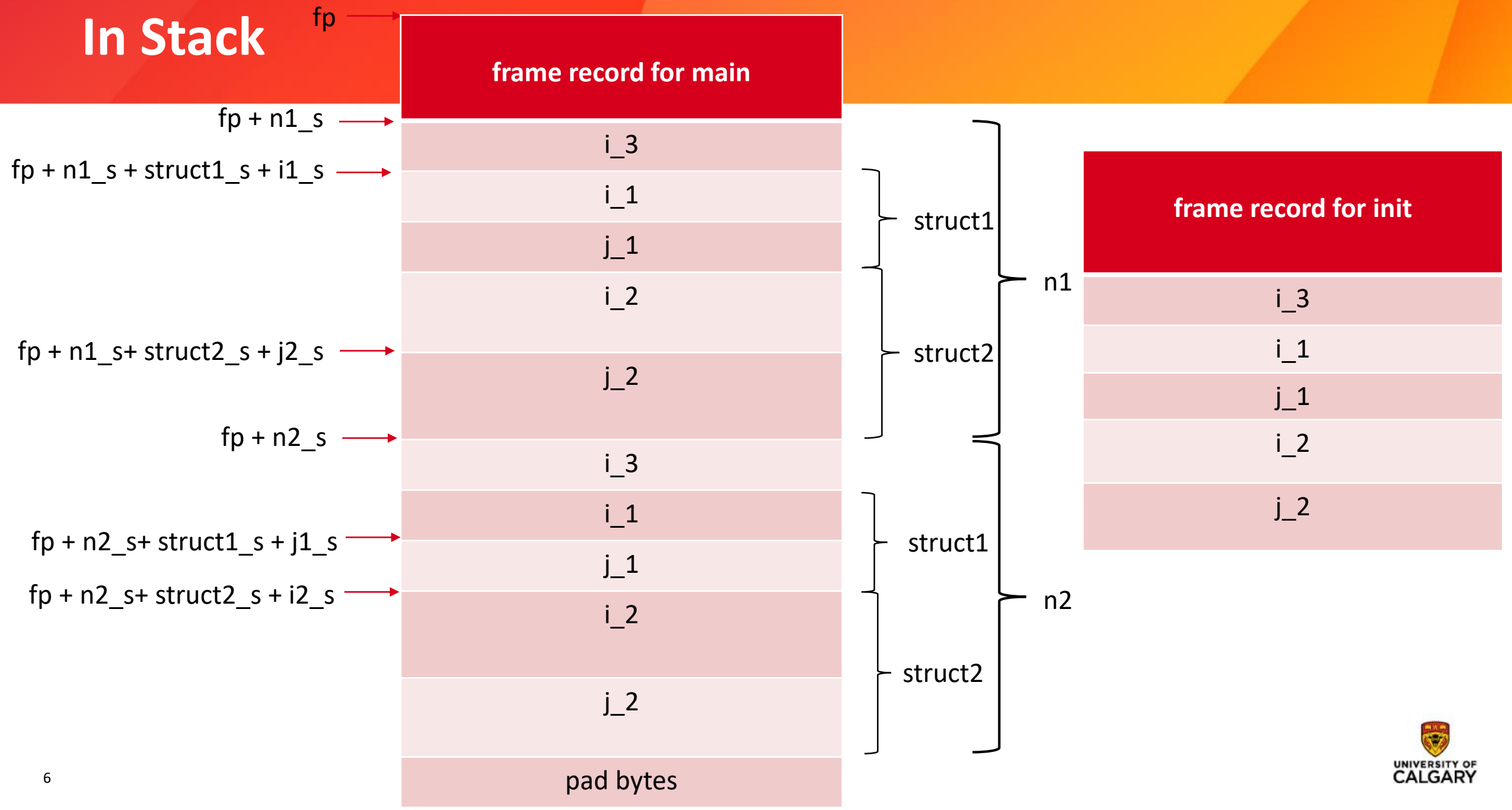
```
int main()
{
        struct nested_struct n1, n2;
        n1 = init();
        n2 = init();
        return 0;
}
```


UNIVERSITY OF CALGARY

```
        //Equates for struct1
        i1_s = 0
        j1_s = 4
        struct1_size = 8
        //Equates for struct2
        i2_s = 0
        j2_s = 8
        struct2_size = 16
        //Equates for nested struct
        i3_s = 0
        struct1_s = 4
        struct2_s = 12
        nested_struct_size = 28
        // function: init
        alloc = -(16 + nested_struct_size) & -16
        dealloc = -alloc
        n_s = 16
        .balign 4
init:   stp         x29, x30, [sp, alloc]!
        mov x29, sp
        // initialize the nested struct
        str wzr, [x29, n_s + i3_s]
        str wzr, [x29, n_s + struct1_s + i1_s]
        str wzr, [x29, n_s + struct1_s + j1_s]
        str xzr, [x29, n_s + struct2_s + i2_s]
        str xzr, [x29, n_s + struct2_s + j2_s]
        //return initialized value by copying it into memory at address in x8
        ldr w9, [x29, n_s + i3_s]
        str w9, [x8, i3_s]
        ldr w9, [x29, n_s + struct1_s + i1_s]
        str w9, [x8, struct1_s + i1_s]
        ldr w9, [x29, n_s + struct1_s + j1_s]
        str w9, [x8, struct2_s + struct2_s]
        ldr x9, [x29, n_s + struct2_s + i2_s]
        str x9, [x8, struct2_s + i2_s]
        ldr x9, [x29, n_s + struct2_s + j2_s]
        str x9, [x8, struct2_s + j2_S]

        ldp  x29, x30, [sp], dealloc
        ret
```

```
// main: create two nested structure n1 and n2
        alloc = -(16 + nested_struct_size*2) & -16
        dealloc = -alloc
        n1_s = 16
        n2_s = n1_s + nested_struct_size
        .global main
main    stp                 x29, x30, [sp, alloc]!
        mov                 x29, sp

        add                 x8, x29, n1_s
        bl                  init
        add                 x8, x29, n2_s
        bl                  init

        ldp                 x29, x30, [sp], dealloc
        ret
```

In Stack



UNIVERSITY OF CALGARY

Pointer Arguments

- the *address* of a variable is passed to the subroutine
- Eg. Swapping numbers

```
void swap(int *num1, int *num2)
{
        register int temp;
        temp = *num1;
        *num1 = *num2
        *num2 = tempnum;
}

int main()
{
        int a = 5, b = 7;
        swap(&a, &b);
}
```

```
        a_size = 4
        b_size = 4
        alloc = -(16 + a_size + b_size)
        dealloc = -alloc
        a_s = 16
        b_s = 20

        .global main
main:   stp     x29, x30, [sp, alloc]!
        mov     x29,sp

        mov     w19, 5
        str     w19, [x29, a_s]
        mov     w20, 7
        str     w20, [x29, b_s]

        add     x0, x29, a_s
        add     x1, x29, b_s
        bl      swap

        ldp     x29, x30, [sp], 16
        ret

define(temp_r, w9)

swap:   stp     x29, x30, [sp, -16]!
        mov     x29, sp

        ldr     temp_r, [x0]
        ldr     w10, [x1]
        str     w10, [x0]
        str     temp_r, [x1]

        ldp     x29, x30, [sp], 16
        ret
```


UNIVERSITY OF CALGARY

Print & modify struct using pointer as argument

- print struct

```
void  printstruct(nested_struct *n )
{
        printf("i_3 = %d, i_1 = %d, j_1 = %d, i_2 = %d",
                n->i_3,
                n->struct1.i_1,
                n->struct1.j_1,
                n->struct2.i_2,
                n->struct2.j_2);
}
```

n->struct1.i_1 is equivalent to (*n).struct1.i_1
n->i_3 is equivalent to (*n).i_3

“n->“ gets the member from the struct that n points to

```
//Equates for struct1
i1_s = 0
j1_s = 4
struct1_size = 8
//Equates for struct2
i2_s = 0
j2_s = 8
struct2_size = 16
//Equates for nested struct
i3_s = 0
struct1_s = 4
struct2_s = 12
nested_struct_size = 28
//function: printstruct
fmtprint  .string   "i_3 = %d, i_1 = %d, j_1 = %d, i_2 = %d, j_2 = %d\n"
printstruct:
        stp                 x29, x30, [sp, -16]!
        mov                 x29, sp
        //call printf
        ldr                 w1, [x0, i3_s]
        ldr                 w2, [x0, struct1_s + i1_s]
        ldr                 w3, [x0, struct1_s + j1_s]
        ldr                 x4, [x0, struct2_s + i2_s]
        ldr                 x5, [x0, struct2_s + j2_s]
        adrp                x0, fmtprint
        add                 x0, x0, :lo12:fmtprint
        bl printf
        //return
        ldp                 x29, x30, [sp], 16
        ret
```


UNIVERSITY OF CALGARY

Print & modify struct using pointer as argument

- modify struct

```
void modifystruct(nested_struct * n, int
i_3, int i_1, int j_1, long int i_2,
long int j_2)
{
        n->i_3 = i_3;
        n->struct1.i_1 = i_1;
        n->struct1.j_1 = j_1;
        n->struct2.i_2 = i_2;
        n->struct2.j_2 = j_2;
}
```

```
modifystruct:
        stp     x29, x30, [sp, -16]!
        mov     x29, sp
        //modify i_3
        ldr     w9, [x0, i3_s]
        mov     w9, w1
        str     w9, [x0, i3_s]
        //struct1
        ldr     w9, [x0, struct1_s + i1_s]
        mov     w9, w2
        str     w9, x0, struct1_s + i1_s]
        ldr     w9, [x0, struct1_s + j1_s]
        mov     w9, w3
        str     w9, x0, struct1_s + j1_s]
        //struct2
        ldr     x9, [x0, struct2_s + i2_s]
        mov     x9, x4
        str     x9, x0, struct2_s + i2_s]
        ldr     x9, [x0, struct2_s + j2_s]
        mov     x9, x5
        str     x9, x0, struct2_s + j2_s]
        //return
        ldp     x29, x30, [sp], 16
        ret
```


UNIVERSITY OF CALGARY

coding pratice

- try to translate c code *nestedstruct.c* into assembly
- solution *nestedstruct.s*

UNIVERSITY OF CALGARY