

ICS Homework 10

December 4, 2020

1 Structs and Unions

Please answer the following questions according to the definition of heterogeneous data structures. Assume it is on an x86-64 machine.

```
struct node_t {
    char type;
    union data_t {
        struct {
            long lsn;
            char loaded;
            struct node_t **child_cache;
            short keys[7];
            long children[8];
        } intern;
        struct {
            struct node_t* (*split) (int);
            short keys[7];
            char values[32];
        } leaf;
    } data;
    char status;
} node;

union data_t *data = &(node.data);
```

This declaration illustrates that structures can be embedded within unions.

1. Fill in the following blocks.(please represent address with **Hex**)

Representation	Value
sizeof(node)	120
sizeof(node.data)	104
sizeof(node.data.leaf)	56
sizeof(node.data.intern)	104
&node	0x601060
&(node.data)	0x601068
&(data->intern.loaded)	0x601070
&(data->intern.children)	0x601090
&(data->leaf.keys)	0x601070
&(data->leaf.values)	0x60107e
&(node.status)	0x6010d0

2. If you can rearrange the declarations in the **struct intern**, how many bytes of memory can you save in **struct intern** compared to the original declaration under x86-64?

Answer: 8 bytes

2 Arrays

Assume we have an array **int a[100]**, the base address is stored in **%rax**. And we have a variable **long i**, stored in **%rbx**. Please write the corresponding instructions to get the value of the expression and store it to **%rcx**:

Instruction	Expression
movl (%rax), %ecx	a[0]
movl 40(%rax), %ecx	a[10]
movl (%rax, %rbx, 4), %ecx	a[i]
movl -28(%rax, %rbx, 4), %ecx	*(a+i-7)
leaq (%rax, %rbx, 4), %rcx	&a[i]
leaq 20(%rax, %rbx, 4), %rcx	&a[i + 5]

3 Nested Array

1. Assume we have to functions:

```
iint array[10][20][30];

int g(int *p, long i, long j, long k) {
    return p[i*600+j*30+k+663];
}
```

```
int foo(long i, long j, long k) {  
    return g(&array[-1][-2][-3], i, j, k);  
}
```

Please fill the blank so that foo will return array[i][j][k];