CSE 30: Computer Organization and Systems Programming

Lecture 14: Midterm review

Diba Mirza
University of California, San Diego
Q2: More fun with pointers
For the C code below make the following assumptions
   i. The base address of array is 0x10000000
   ii. The byte ordering is Little Endian
   iii. int is 4 bytes

```c
int array[5] = {-1,1,2,3,4,5};
int main()
{
    char *ptr1 = (((char *) array) + 5);
    char val;
    int *ptr2 = array + 3;
    val = *ptr1;
    return 0;
}
```

- Show the byte level representation of ‘array’
- What is the value of *ptr2 and val before main returns?
Compile by hand

• $g = h + A[8]$
Accessing arrays

```c
void foo (int *p, int size) {
    *p = 0;
    *(p+1) = 0;
    *(p+size-1) = 0;
}
```

In ARM:
Data Transfer Instructions

How would the code change if int was replaced by short?

In ARM:

```c
void swap (int *x, int *y){
    int tmp =*x;
    *x = *y;
    *y = tmp;
}
```
struct Node {
    int value;
    struct Node *next;
};

Q: Assume the linked list has already been created, what do the following expressions evaluate to?

1. head->value  
   A. 10

2. head->next->value  
   B. 5

3. head->next->next->value  
   C. 15

4. head->next->next->next->value  
   D. NULL

5. head->next->next->next->next->value  
   E. Run time error
Q: Write ARM Assembly for each of the following. Assume head is in r0

- head->value=100;
- head->next->value=100;

```c
struct Node {
    int value;
    struct Node *next;
};
```
Let’s look at an example of using structures, pointers, `malloc()`, and `free()` to implement a linked list.

typedef struct Node node;

struct Node {
    int *int;
    _____ next;
};

Q: What should be the data type of the variable ‘next’?

A. struct Node
B. Node
C. node
D. node *
Adding a node to the list

```c
node *list_add(node* head, int new_value)
{
    /* Add a new node on the head of the list,
    Populate it with new_value and return the new head */
}
```
What should Line 5 be to achieve the pointer diagram (below)?

A. `new_node->next = head;`
B. `next = head;`
C. `head = new_node;`
D. `new_node->next = *head;`
node *list_add(node* head, int new_value)
{
    node *new_node=(node*) malloc(sizeof(node));
    if (!new_node)
        return NULL;
    new_node->value = new_value;
    new_node->next = head;
    return _____________;
}
node *list_add(node* head, int new_value)
{
    node *new_node=(node*) malloc(sizeof(node));
    if (!new_node)
        return NULL;
    new_node->value = new_value;
    new_node->next = head;
    return new_node;
}

Translate to ARM assembly
While loops

while (a<0)
    a++;
For loops

for (i=0; i<10; i++){
    a++;
    b--;
}