Basic Array Example

COS1004 Lecture 9.4

Iterating though an array

- Consider this psuedo code – what is it doing ?

```
numarray[10]: {3, 2, 5, 3, 5, 6, 1, 2, 4, 9}
sum = 0
i = 0
while(i < 10)do
     sum = sum + numarray[i]
      i = i + 1;
end while
```

Iterating though an array

- Consider this psuedo code – what is it doing ?

```
numarray[10]: {3, 2, 5, 3, 5, 6, 1, 2, 4, 9}
sum = 0
i = 0
while(i < 10)do
     sum = sum + numarray[i]
      i = i + 1;
end while
```

It’s defining an array of 10 integers and summing the values

Iterating though an array

- Lets do this in Assembly code
- First, define the array with some labels:

```
arraysize: 40
numarray: .Word 3
 2
 5
 3
 5
 6
 1
 2
 4
 9
```

Iterating though an array

- Lets do this in Assembly code
- Next, initialise some registers:

```
MOV R0, #numarray
MOV R1, #0     // index
MOV R2, #0      // sum
HALT
arraysize: 40
numarray: .Word 3  2  5 3 5  6  1 2 4 9   // <-
these should be one per line in actual code
```

Iterating though an array

- Lets do this in Assembly code
- Next, setup a label and conditional branch for looping

```
MOV R0, #numarray
MOV R1, #0    // index
MOV R2, #0     // sum
arrayloop:
CMP R1, #arraysize
BLT arrayloop
HALT
arraysize: 40        // 10 * 4 bytes
numarray: .Word 3  2  5 3 5  6  1 2 4 9   // <- these should be one per line in actual code
```

Iterating though an array

- Lets do this in Assembly code
- Next, insert code to access each item in array and increment index

```
MOV R0, #numarray
MOV R1, #0    // index
MOV R2, #0          // sum
arrayloop:
LDR R3, [R0 + R1]           // access array item at current index (R1)
ADD R1, R1, #4                      // increment index to next 32 bit word
CMP R1, #arraysize
BLT arrayloop
HALT
arraysize: 40   // 10 * 4 bytes
numarray: .Word 3  2  5 3 5  6  1 2 4 9   // <- these should be one per line in actual code
```

Iterating though an array

- Lets do this in Assembly code
- Next, insert code to access each item in array and increment index

```
MOV R0, #numarray
MOV R1, #0    // index
MOV R2, #0       // sum
arrayloop:
LDR R3, [R0 + R1]                // access array item at current index (R1)
ADD R1, R1, #4                    // increment index to next 32 bit word
CMP R1, #arraysize
BLT arrayloop
HALT
arraysize: 40  // 10 * 4 bytes
numarray: .Word 3 2 5 3 5 6 1 2 4 9   // <- these should be one per line in actual code
```

Iterating though an array

- Lets do this in Assembly code
- Finally, add each value to the accumulating “sum”

```
MOV R0, #numarray
MOV R1, #0    // index
MOV R2, #0    // sum accumulator
arrayloop:
LDR R3, [R0 + R1]                // access array item at current index (R1)
ADD R2, R2, R3                   // add value to accumulator (R2)
ADD R1, R1, #4                   // increment index to next 32 bit word
CMP R1, #arraysize
BLT arrayloop
HALT
arraysize: 40            // 10 * 4 bytes
numarray: .Word 3 2 5 3 5 6 1 2 4 9 <- these should be one per line in actual code
```