

RS2318

Dynamic Arrays

10/13/2016

1) 1D dynamic array

H.L.L ~~*~~ for Creating ↗

2) 2D dynamic array

H.L.L ~~*~~ = on assembly

int size;

i) int *Arr;

Arr = ~~new~~ ~~*~~ new int [size];

Assume size = 15

Compiler enters into a table

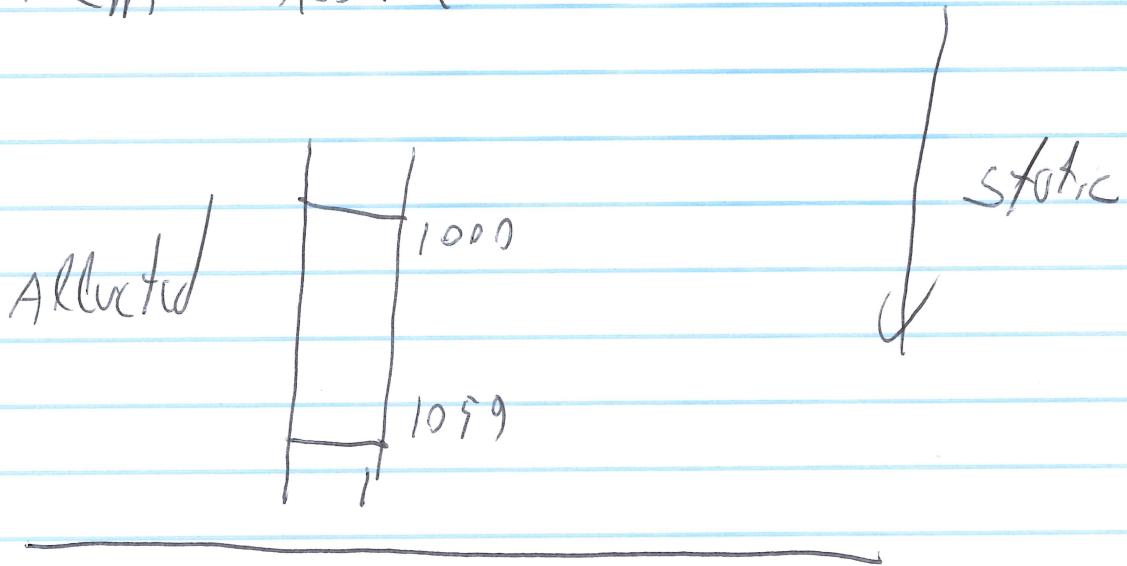
Link
1000 → | Arr pointer to ints

0515 for fn byte

2

Linker Allocates 60 bytes in

Mem Assume



Dynamic Mem

int * Arr;

Arr = * new int[15];

static

ARR parent pointer to ints

Dynamic time JS provides

60 bytes starting from base

3

Execution of this instruction

Allocates the space

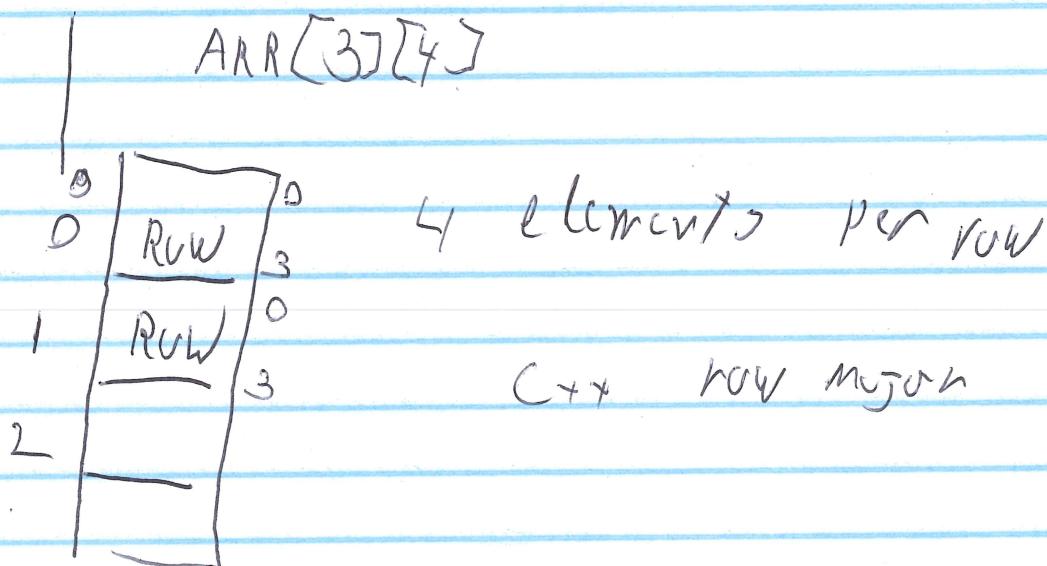
of 5 returns base

int *Arr;

Arr = (*int) malloc (15 * sizeof(int))

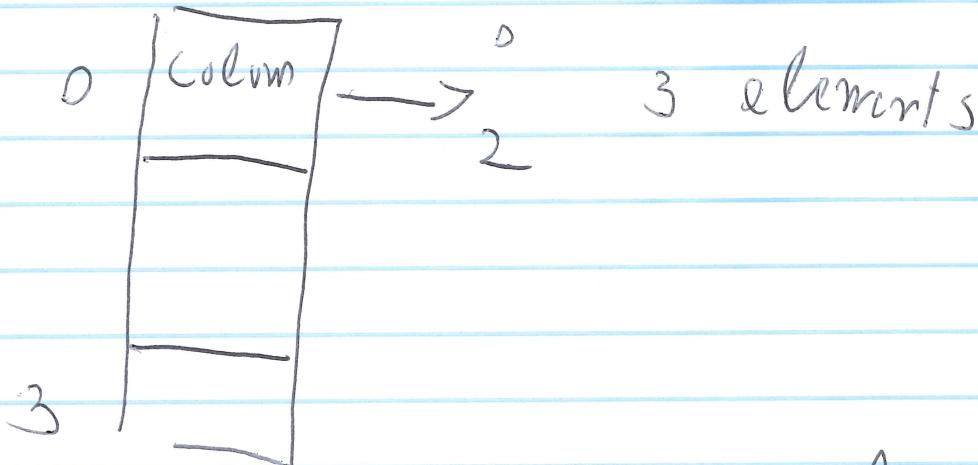
15 * 4 (60)

HLL 2-D Dynamic Array



4

Column Major (3×4)



Array of

`int **Arr;` / pointers
Arr = new int *[RowCount];

for (int i = 0; i < RowCount; i++)

 Arr[i] = new int [ColCount]

 \ 4

Arr = new int[15]

slurh sys call 9

\$0 - bytes requested
\$0 returns the pointer

5

start \$0, number of bytes

requested

\Rightarrow in ~~\$10~~ \rightarrow Allocates bytes

\Rightarrow pointer to the first byte

or null

L: \$10, 9

L: \$0, 60

syscall ~~int~~ * A;

A = new int[60];

A[0] = 5;

INT * A;

A = NEW int [15];

A[0] = 5;

6

Jota
Align 2
Space 4

.text

Li \$V0, \$99

LI \$A0, 60 60 byte is
ints

syscall \$V0 contains base

mov \$S0, \$V0 \$S0 is base

Li \$T0, 5

sw \$T0, 8(\$S0)

~~addi \$T0, \$T0, 8~~

~~addi \$S0, \$S0, 7~~

sw \$T0, 0(\$S0)

La \$T1, A] pointer
sw \$V0, 0(\$T1)] is in A

7

.text

li \$v0, 9

li \$t0, 12

sgscole

\$v0 will be
a pointer to 3 printen

mov \$sp, \$v0

i li \$t0, 0 while i < 3

lwrd li \$t1, 3

while: bge \$t0, \$t1, endwhile

li \$t0, 9 lw \$s1, 0(\$s0)

li \$t0, 16 addi \$s0, \$s0, 4

sgscole

mvn \$s1, \$v0

b while
endwhile

8

Creating a reentrant ^{le} single dimensional
array (ints)

No checking for bounds

Set o is int array

mov \$50, \$SP

Assume \$SP

15 ints
+ pointer

subi \$SP, \$SP, 64

points to
next avail
stack grows down

pointer

→ sw \$50, 0(\$50)

FFFF

Array

redundant

push

Now how to generate a 2-d

Dynamic on Heap (System)

And on stack;