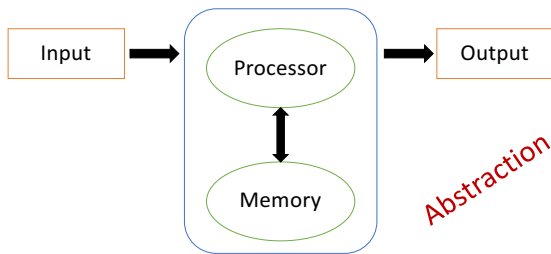


Memory Diagrams

Michael R. Nowak
Texas A&M University
Sept. 12, 2017

Slides created by J. Michael Moore

Simplified Model



Memory

- Usually stored in RAM
- Composed of ones and zeroes
- Address for each byte (group of 8 bits)
 - Physical Address
 - Logical Address
 - Starts at zero

Memory Layout



Stack and heap
grow toward each other.

Memory Diagram

- It is not helpful for us to refer to specific addresses. We tend to think symbolically about the data.
- For example we think about x times x
 - rather than thinking x refers to a memory address and if we get the value held at that address and multiply it by that value...
- Memory diagrams allow us to think about the variables we are using in a program without having to worry about specific memory addresses.
- They can also help us do 'hand execution' of the code.

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

Program

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

Set up memory diagram...
An area for the stack and
an area for identifiers...

identifier | stack

output

Program

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

Set up area to write output...

identifier | stack

output

Program

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

We will identify the
name of the function
(only main for now).

main

identifier | stack

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Add variable identifiers as we encounter them...

Note: We don't know what value rank holds. It could be any random values for its bits. When we initialize we set it to a known value!

main	rank	stack
identifier		

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Set values when assigned...

Note: This is first value assigned so it is initialization.

main	rank	stack
identifier	15	

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

main	classSize	rank	stack
identifier	35	15	

output

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

main	stack
score1	82.45
classSize	35
rank	15
identifier	

output

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

main	stack
score2	95.25
score1	82.45
classSize	35
rank	15
identifier	

output

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

main	stack
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15
identifier	

output

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

main

identifier	stack
grade	C
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15

output

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

main

When assigned a new value, cross out old value and write in new value.

identifier	stack
grade	C
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15 7

output

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

main

identifier	stack
grade	C B
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15 7

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

name	Michael
grade	C B
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15 7

main

identifier stack

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Name: Michael

Now start output...

Note: endl means go to a new line

name	Michael
grade	C B
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15 7

main

identifier stack

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Name: Michael
Average: 88.85

name	Michael
grade	C B
average	88.85
score2	95.25
score1	82.45
classSize	35
rank	15 7

main

identifier stack

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Name: Michael
Average: 88.85
Rank: 7 of

main	name	Michael
	grade	C B
	average	88.85
	score2	95.25
	score1	82.45
	classSize	35
	rank	15 7
identifier		stack

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Name: Michael
Average: 88.85
Rank: 7 of 35

main	name	Michael
	grade	C B
	average	88.85
	score2	95.25
	score1	82.45
	classSize	35
	rank	15 7
identifier		stack

Program

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int rank = 15;
    int classSize = 35;
    double score1 = 82.45;
    double score2 = 95.25;
    double average = (score1 + score2) / 2;
    char grade = 'C';
    rank = 7;
    grade = 'B';
    string name = "Michael";
    cout << "Name: " << name << endl;
    cout << "Average: " << average << endl;
    cout << "Rank: " << rank << " of ";
    cout << classSize << endl;
    cout << "Grade: " << grade << endl;
}
```

output

Name: Michael
Average: 88.85
Rank: 7 of 35
Grade: B

main	name	Michael
	grade	C B
	average	88.85
	score2	95.25
	score1	82.45
	classSize	35
	rank	15 7
identifier		stack

A tool

- Memory diagrams are only a tool.
- Usually do them on scratch paper.
- On an exam, make legible.
- The next slide is an example of what this example might really look like...

Output

Name: Michel
Average: 88.85
Rank: 7 of 35
Grade: B

Michel	grade	B
	average	88.5
	Score 2	95.25
	Score 1	82.45
	Change	35
Rank		18/7
Number		35
