Virtual Memory

- dedicated address space for a process (program)
- appears to be a large amount of contiguous memory
  - virtual memory addresses
- actual memory can be anywhere (RAM, disk, etc)
  - physical memory addresses

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
	...

Areas of virtual memory

- code (text) segment
  - program instructions
- data segment
  - global variables
  - static variables
  - heap (dynamic memory)
  - typically grows down
- call stack
  - FIFO
  - typically grows up

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
	...

```c
#include <stdio.h>

int x = 3;
char c = 'Q';

int main(){
   int x = 12;
   float f = 3.145f;
}
```

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
	x11b8
	x11bc
	x11c0
	x11c4
	x11c8
	x11ca

```
#include <stdio.h>

int x = 3;
char c = 'Q';

int main(){
   int x = 12;
   float f = 3.145f;
}
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

```
#include <stdio.h>

int x = 3;
char c = 'Q';

int main(){
   int x = 12;
   float f = 3.145f;
}
```

Virtual Memory

	x0000
	x0004
	x0008
0000...00000011 (x)	x000c
0000...00110011 (c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
x404947AE (main's f)	x11b8
0000...00001100 (main's x)	x11bc
	x11c0
	x11c4

```
#include <stdio.h>

int x = 3;
char c = 'Q';

int main(){
   int x = 12;
   float f = 3.145f;
}
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

Address of operator: &

- returns the address in memory of a variable

```
char c = 'Q';

int main(){
    int x = 12;
    float f = 3.145f;
    printf("%p", &x);
    printf("%p", &f);
    printf("%p", &c);
}
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

Dereference operator: *

- grab what is at a given memory location

```
char c = 'Q';

int main(){
    int x = 12;
    float f = 3.145f;
    printf("%d", *(&x));
    printf("%e", *(&f));
    printf("%c", *(&c));
}
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

Pointers

- memory addresses is also valid data

```
char c = 'Q';
char *c_ptr = &c;

int main(){
    printf("%p", &x);
    printf("%c", *c_ptr);
    printf("%c", *(&c));
}
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

Pointers

- memory addresses is also valid data

```
char c = 'Q';
char* c_ptr = &c;

int main(){
    printf("%p", &x);
    printf("%c", *c_ptr);
    printf("%c", *(&c));
}
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

!!DANGER!!

```
char * a, b;

means

char * a;
char b;

and not

char *a;
char *b;
```

Virtual Memory

	x0000
	x0004
	x0008
3 (global x)	x000c
Q (global c)	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
3.145f (local f)	x11b8
12 (local x)	x11bc
	x11c0
	x11c4

Automatic Arrays

```
int array[3];

char c[2] = {'a','b'};

int x[] = {-1,2,3};
```

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
	x11a0
	x11a4
	x11a8
	x11ac
	x11b0
	x11b4
	x11b8
	x11bc
	x11c0
	x11c4

Automatic Arrays

```
int array[3];

char c[2] = {'a','b'};

int x[] = {-1,2,3};
```

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
-1	x11a0
2	x11a4
3	x11a8
'a'	x11ac
'b'	x11b0
?	x11b4
?	x11b8
?	x11bc
	x11c0
	x11c4

Automatic Arrays

```
int main(){
   int array[3];
   char c[2] = {'a','b'};
   int x[] = {1,2,3};

   printf("%p %p %p",
          &array, &c, &x);

}
```

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
-1	x11a0
2	x11a4
3	x11a8
'a'	x11ac
'b'	x11b0
?	x11b4
?	x11b8
?	x11bc
	x11c0
	x11c4

Automatic Arrays

```
int main(){
   int array[3];
   char c[2] = {'a','b'};
   int x[] = {1,2,3};

   printf(sizeof(x));

}
```

Virtual Memory

	x0000
	x0004
	x0008
	x000c
	x0010
	x0014
	...
-1	x11a0
2	x11a4
3	x11a8
'a'	x11ac
'b'	x11b0
?	x11b4
?	x11b8
?	x11bc
	x11c0
	x11c4