University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 220: Computer Systems & Programming

Implementing Dynamic Allocation
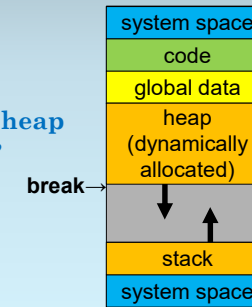

Dynamic Allocation Interacts with the OS

Recall our canonical memory map.

The **address**
- **after the end of the heap**
- is called **the "break."**

To change the break, make a system call.

system space
code
global data
heap (dynamically allocated)
break→
stack
system space

---

sbrk Adjusts the Address of the Break

In Linux, for example, the call is

```
void *sbrk (intptr_t increment);
```

Calling **sbrk** requests that
- the break be changed by adding **increment**,
- and returns the address of the previous break (or **((void*)-1)** on failure).

One can grow or shrink the heap with **sbrk**.

---

intptr_t is Needed to Hold the **sbrk** Argument

```
void *sbrk (intptr_t increment);
```

But what's an intptr_t?

An integer large enough to hold a pointer.

These became important with 64-bit address spaces.

An **int** can no longer hold a pointer!