Introduction to Memory

CS 241
Sept. 6, 2013
MP1
void main() {
    int i, j;
    void *s1 = &i,
        *s2 = &j;
    void *h1 = malloc(sizeof(int)),
        *h2 = malloc(sizeof(int));

    printf("s1: %p\ns2: %p\n", s1, s2);
    printf("h1: %p\nh2: %p\n", h1, h2);
}
Program #1
Address Space

Observation:

Implication:
Virtual Memory

• You have discovered a result of virtual memory.
  – All modern OSs provide a level of indirection between processes and the underlying RAM.
  – What if we didn’t have virtual memory?
Alternative Option

Segmentation
• Each process owns a segment of the system’s RAM.

Example Allocations:
1. P1: 3 MB
2. P2: 5 MB
3. P3: 2 MB
4. P2 is free’d
5. P4: 4 MB
6. P5: 5 MB
7. P1: Increase to 5 MB
Segmentation

• Problems:
  1.
  2.

• Advantages?
Virtual Memory

• **Virtual memory** is derived from segmentation, with two major differences:
  
  – The entire memory space is divided up into fixed sized segments called **pages**.
  
  – Each and **every process** on an OS has its own **page table** to translate between “**virtual addresses**” (used in user-space) and “**physical addresses**” (used in kernel-space, the address on the physical RAM).
p = malloc(1 MB);

q = malloc(2.1 MB);

free(p);

r = malloc(2 MB);