CptS 260
Introduction to Computer Architecture
Unit 6: Memory (Part 2)

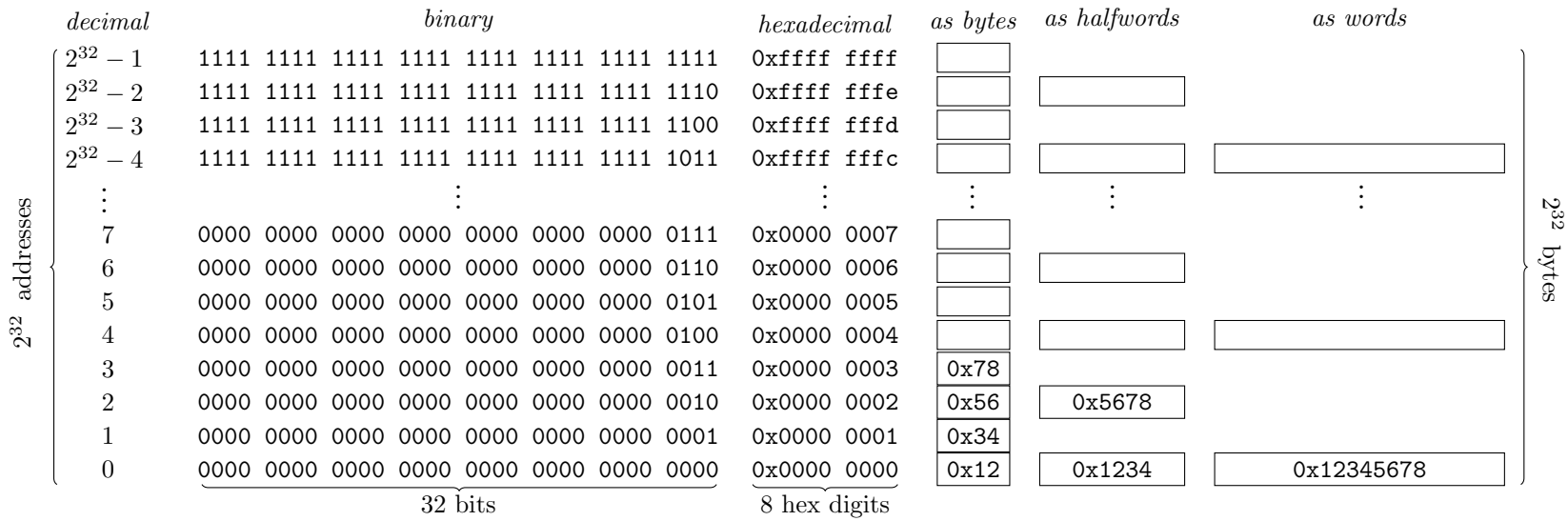
Bob Lewis

School of Engineering and Applied Sciences
Washington State University

Fall, 2021

The MIPS32 Address Space (Review)

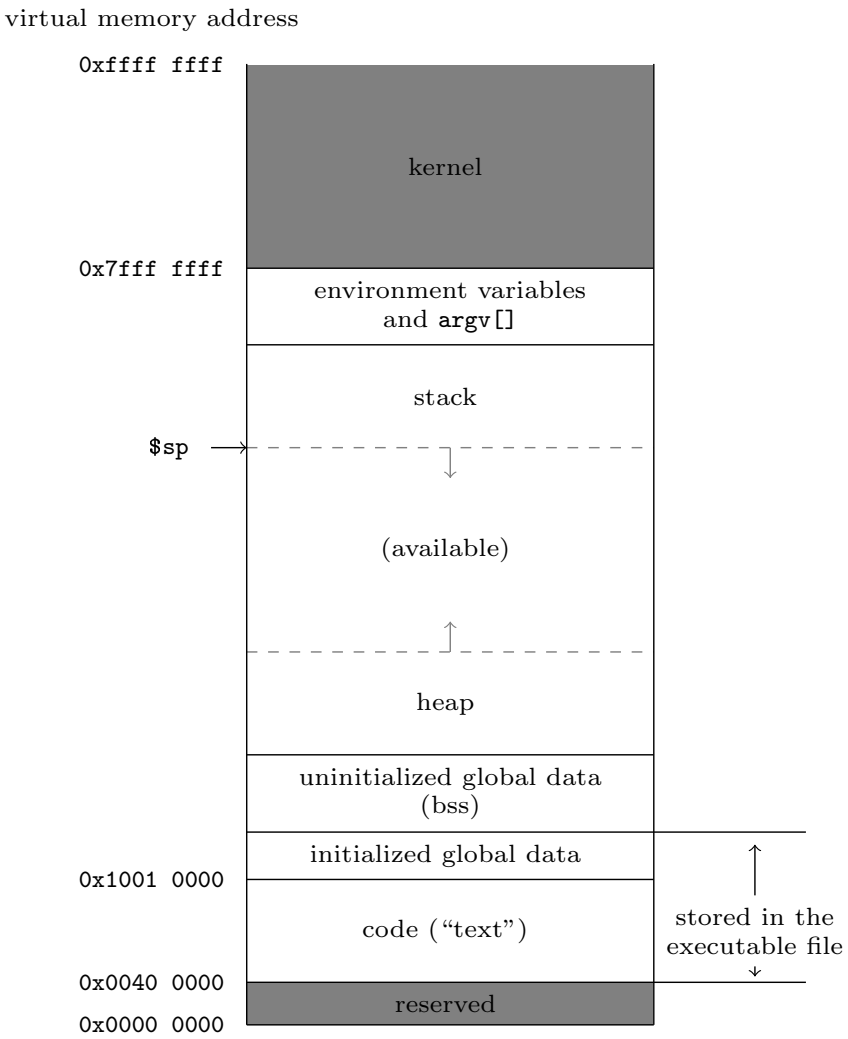
This diagram shows the layout of physical memory in MIPS32 CPU's address space:



How Does the MIPS Actually *Use* Virtual Memory?

- Just because you've got a 32-bit virtual address space doesn't mean your program can use all of it.
- The kernel is not a separate process, just like Superman is not a separate person from Clark Kent.
- Your program *becomes* the kernel by entering supervisor mode (with the "`syscall 0x20`" exception).
  - This jumps to the handler, which can then access kernel memory.
- VM protects your program from other programs, but it also needs to protect the kernel (especially kernel memory) from *you*!
- Preventing unauthorized access to this memory is a major cybersecurity task.

Partitioning Virtual Memory



- This is how Linux allocates your virtual address space.
- We'll explain that "reserved" region in the I/O Unit.
- (see the *vm_addresses* demo)

Page Tables Revisited

In addition to the **V** ("valid") bit (which means the same in a page table as it does in a cache), the memory manger hardware may also allow these bits in each entry (not shown in the textbook):

- **RW**: The page is writeable
  Q: What memory regions do we not want to write to?
- **USER**: The page is accessible from user space
  Q: What memory regions do we not want the (mere mortal) user to read from or write to?
- **DIRTY**: The page has been written to so it needs to be written to RAM when replaced.
- **ACCESSED**: The page has been read or written recently. This allows a simple approximation to LRU for page replacement.
- **EXECUTABLE**: The PC can be set to addresses on the page.
  Q: What memory regions should this apply to?

Caching and VM on the MIPS32

The MIPS32 architecture is configurable for the customer:

- The cache size can be 0 (no caching), 16kB, 32kB, or 64kB and is 4-way set associative.
- The MIPS architecture allows variable page sizes of 4kB, 8kB, 16kB, 32kB, or 64kB.
- Knowing these sizes allows us to reduce run time and make more efficient use of memory.

Example: Taking Advantage of Cache Coherence I

Consider two implementations of a simple "Name" data structure:

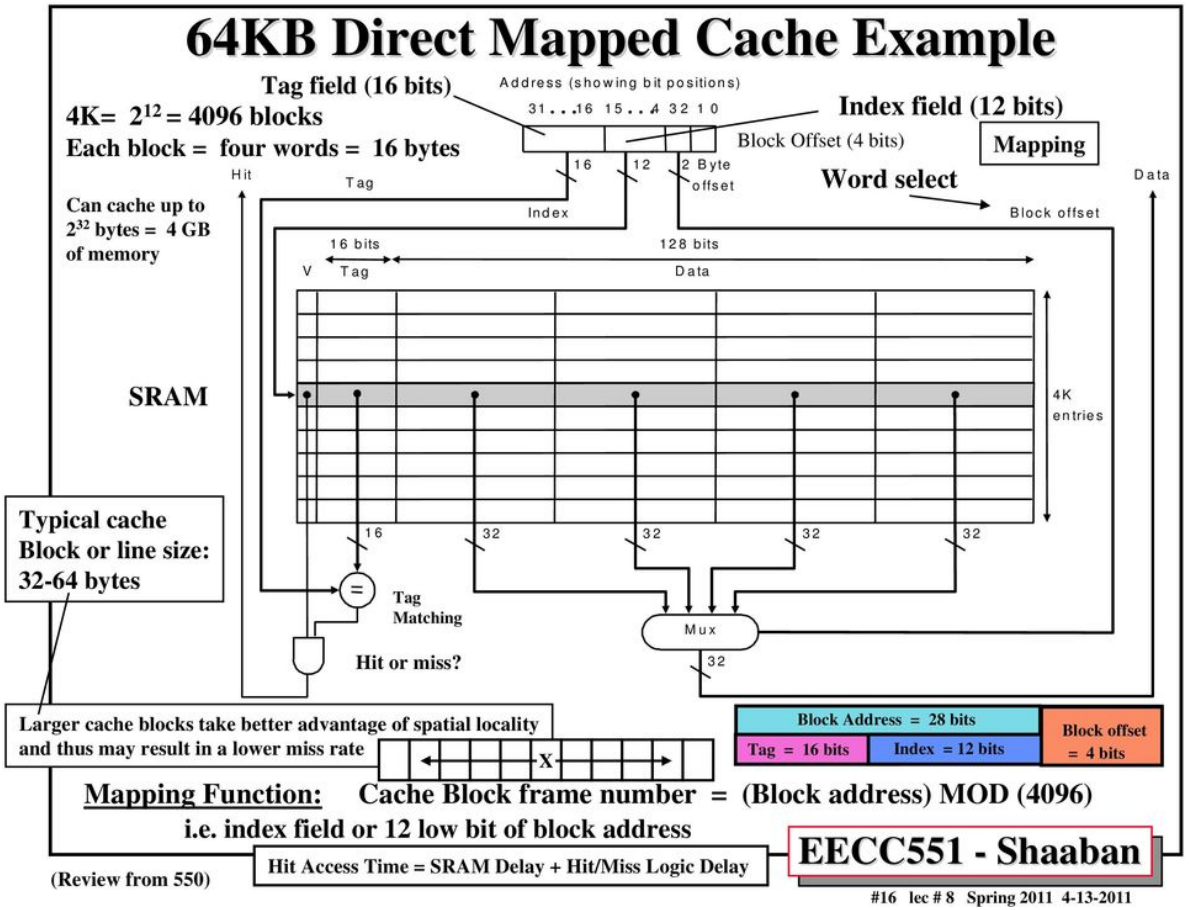
```
struct NameA {
    char first[32];
    char last[32];
};
```

```
struct NameB {
    char *first;
    char *last;
};
```

(`first` and `last` would be allocated on the heap as needed)

- Q: Which of these has is more flexible (i.e. fewer restrictions)?
- Q: Which would be more likely to be directly read from or written to a file or database?
- Q: Which of these typically uses less memory space?
- Q: Given a pointer to either structure, how many memory accesses are required to load the first byte in either name?
- Which of these takes more advantage of spatial locality? ...

Example: Taking Advantage of Cache Coherence II



- How would you rate the two data structure designs for this caching scheme?
- How could the efficiency of `NameA` be improved by a change in *requirements*?