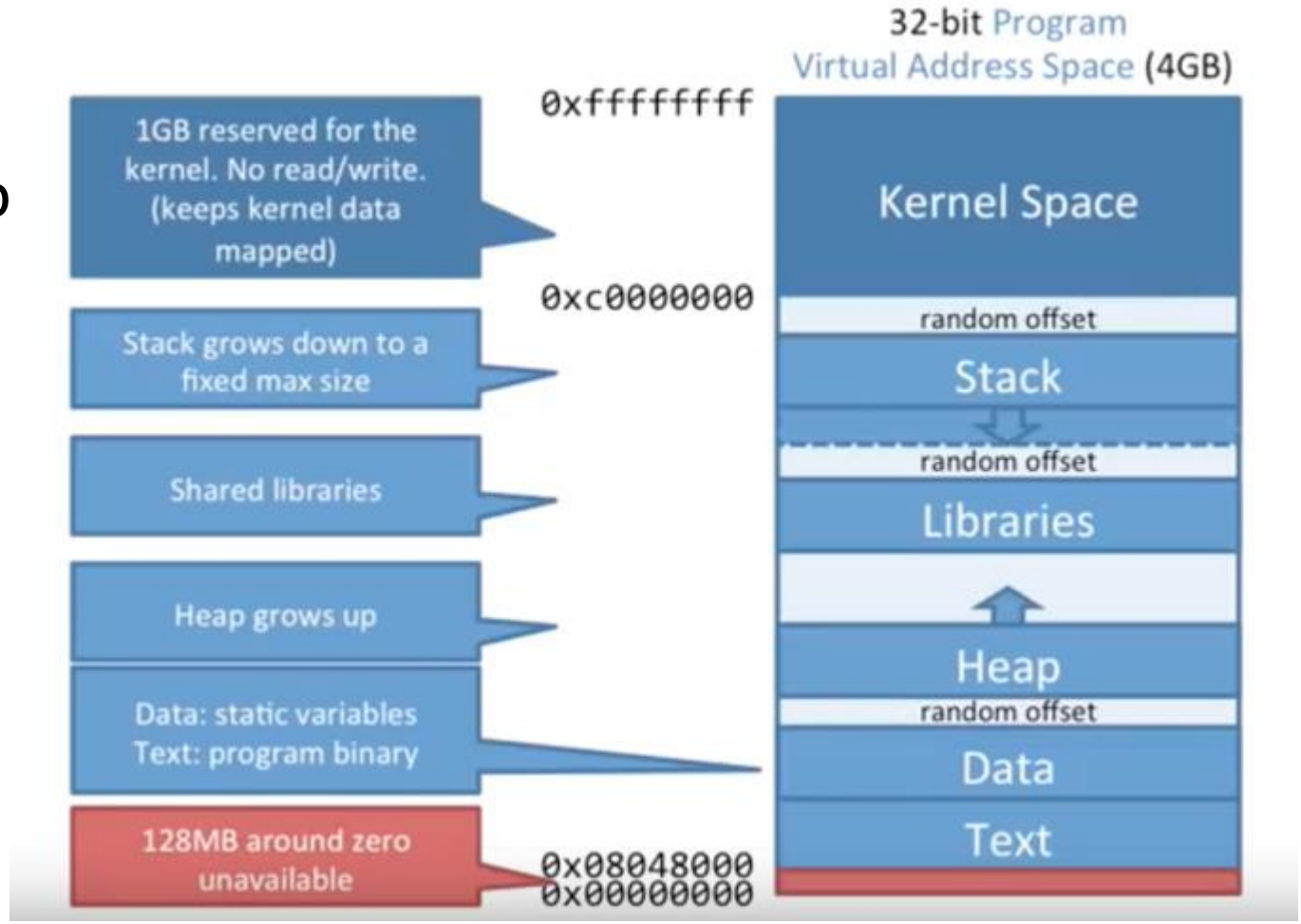
CS 361 – Lab 7

Monday, March 11th, 2019.

Virtual Memory

Why Virtual Memory ?

- Not enough space to accommodate all the programs in physical memory
- Memory fragmentation
- Memory protection from other pro



Program Address space in Linux.

Paging

- Split virtual memory into pages
- Split physical memory into frames
- Map pages to frames
- Mapping enables
  - Pages to be held in physical memory (a valid mapping)
  - Pages to be stored on Disk
  - Permission to be given on a per page basis and checked on every update
- How ?
  - Page Tables….

Page Tables

- Maps pages to frame, need one entry for each page
- Consider a 4KB page, 32-bit virtual address space
  - Need one entry per page
  - Offset (12 bits) is the same in virtual and physical address
  - Map only the high order 20 bits from virtual to physicall
  - Low order 12 bits are the same in virtual and physicall
  - Since only 20 bits are translated each entry has 12 bits for other information.
- Problem, 1 Million entries, 4 bytes each, 4 MB space of Page Tabler program.
- Solution ?
  - Multi Level Page Tables.

Multi Level Page Tables.

- Added level of indirection
  - Each non-null entry in level 1, maps to level 2 page table.
  - 20 bits in the previous example, can be further partitioned.
    - 10 bits for offset for right bits gives 1024 entries in each page table
    - Left 10 bits used for level 1 page table.
  - Advantages:
    - Level 2 page table can be stored in disk
    - Level 1 has to be always in RAM.

- What will be the minimum memory consumed by a program with the above arrangement ?