13. The Abstraction: Address Space

Operating System: Three Easy Pieces

Memory Virtualization

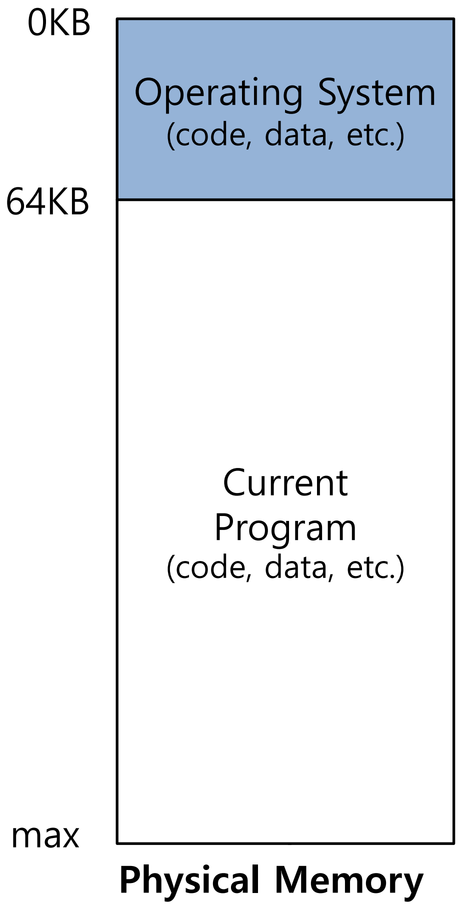
- What is **memory virtualization**?
  - OS virtualizes its physical memory.
  - OS provides an illusion memory space per each process.
  - It seems to be seen like each process uses the whole memory .

Benefit of Memory Virtualization

- Ease of use in programming
- Memory efficiency in terms of time and space
- The guarantee of isolation for processes as well as OS
  - Protection from errant accesses of other processes

OS in The Early System

- Load only one process in memory.
  - Poor utilization and efficiency

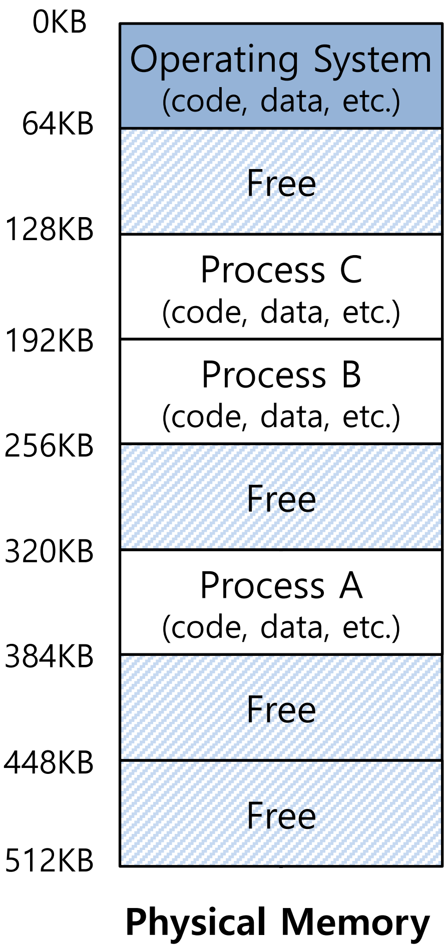


Perhaps in the future too?: Uni-kernels)

Multiprogramming and Time Sharing

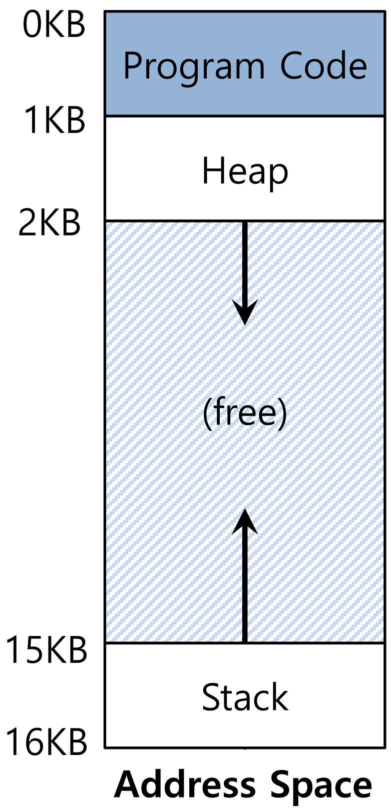
- **Load multiple processes** in memory.
  - Execute one for a short while.
  - Switch processes between them in memory.
  - Increase utilization and efficiency.

- Cause an important **protection issue**.
  - Errant memory accesses from other processes



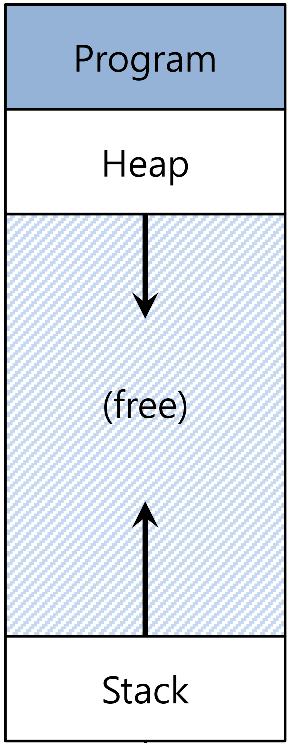
Address Space

- OS creates an **abstraction** of physical memory.
  - The address space contains all about a running process.
  - That is consist of program code, heap, stack and etc.



Address Space(Cont.)

- Program
  - Where instructions and static data live
- Heap
  - Dynamically allocate memory.
    - `malloc/free` in C language
    - `new/delete` in object-oriented language
  - Implicitly handled in (memory) managed languages
- Stack
  - Store return addresses or values.
  - Contain local variables arguments to routines.
  - Implicitly handled in HLL



Address Space

Virtual Address

- **Every address** in a running program is virtual.
  - OS provides the “mechanism” to translate the virtual address to physical address
  - HW assists to make such translation “painless”

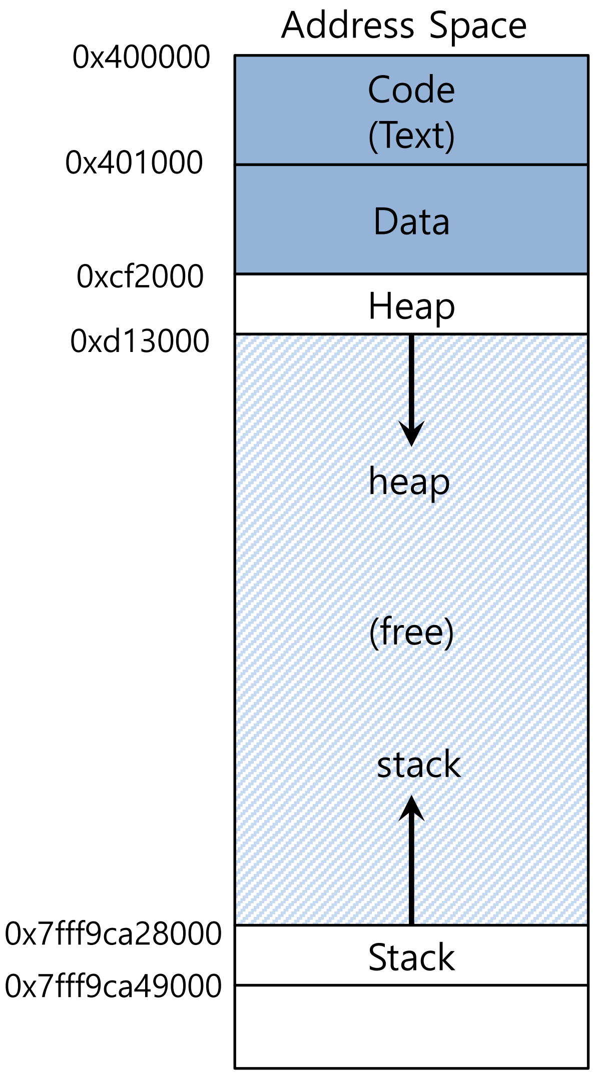
```
#include <stdio.h>
#include <stdlib.h>
int global=1;
int main(int argc, char *argv[])
{
    int x = 3;
    printf("location of code  : %p\n", (void *) main);
    printf("location of data  : %p\n", (void *) &global);
    printf("location of heap  : %p\n", (void *) malloc(1));
    printf("location of stack : %p\n", (void *) &x);

    return x;
}
```

A simple program that prints out addresses

Virtual Address(Cont.)

- The output in 64-bit Linux machine

```
location of code  : 0x40057d
location of data  : 0x401010
location of heap  : 0xcf2010
location of stack : 0x7fff9ca45fcc
```



Goals of VM

- Transparency
  - VM should be invisible to running program
- Efficiency
  - Minimize overhead in terms of speed and space
- Protection
  - Isolate processes (and OS itself) [but allowing selective “communication”]

- Disclaimer: Disclaimer: This lecture slide set is used in AOS course at University of Cantabria. Was initially developed for Operating System course in Computer Science Dept. at Hanyang University. This lecture slide set is for OSTEP book written by Remzi and Andrea Arpaci-Dusseau (at University of Wisconsin)