CSCE 451/851
Operating Systems Principles

Memory Management Basics

Steve Goddard
goddard@cse.unl.edu

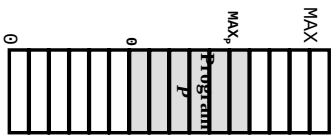
http://www.cse.unl.edu/~goddard/Courses/CSCE451

Memory Layout

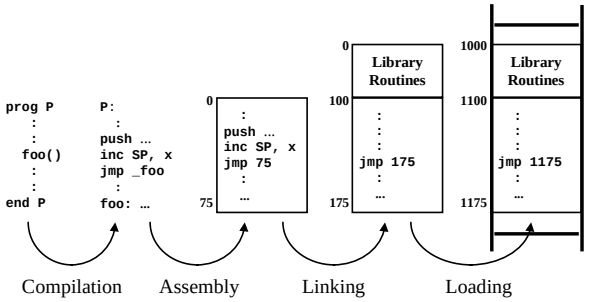
Address spaces

- Memory — a linear array of bytes
  - » *Physical address space* — that supported by the hardware
    - ❖ starting at address 0, going to address `MAX`
  - » *Logical address space* — a process's view of memory
    - ❖ starting at address 0, going to address MAX_p
- But where do addresses come from?

```
MOV r0, 0xffa620e
```

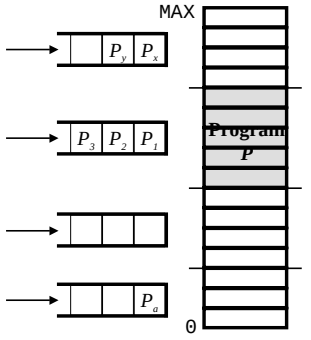


Address Generation
The compilation pipeline



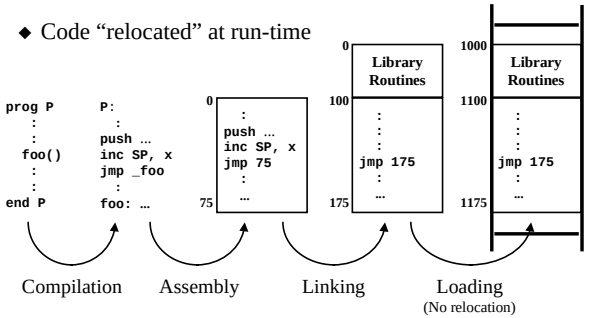
Simple Memory Management Schemes
Fixed-sized partitions

- Two variations:
  - » Jobs queue for partitions
    - ❖ addressing done by the linker/loader
  - » Jobs queue for memory
    - ❖ *dynamic relocation* required
- Scheduling
  - » maximize performance
  - » maximize memory utilization

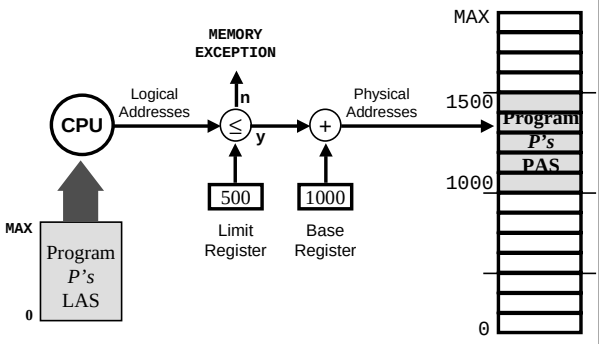


Dynamic Program Relocation
Program compilation

- Code "relocated" at run-time

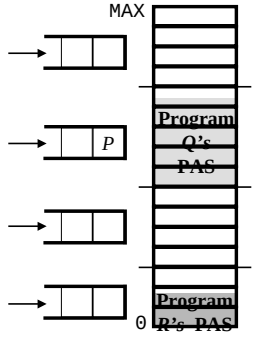


Dynamic Program Relocation
Base + Limit registers



Memory Management Issues
Fragmentation

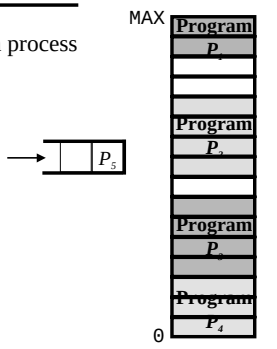
- Internal fragmentation
- External fragmentation



7

Simple Memory Management Schemes
Variable-sized partitions

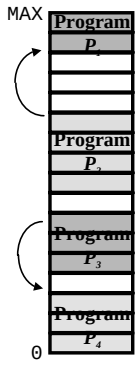
- Allocate a partition when a process is admitted into the system
- Keep track of...
  - » full-blocks
  - » empty-blocks (holes)
- Allocation strategies
  - » first-fit
  - » best-fit
  - » worst-fit



8

Variable-Sized Partitions
Eliminating Fragmentation

- Compaction
  - » relocate programs to coalesce holes
- Swapping
  - » preempt processes & reclaim their memory

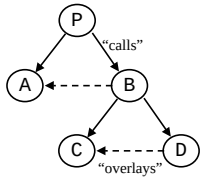


9

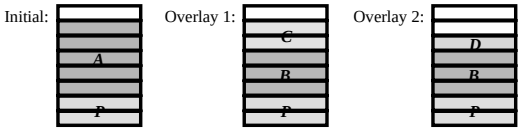
Executing programs larger than physical memory
Overlays

- Program call graph

```
prog P          func A()        func B()
  :               :               :
  A()             :               C()
  B()           end A             D()
  :                               :
end P                           end B
```



- Memory layout



10