Memory Layout

Address spaces

- Memory — a linear array of bytes supported by the hardware
- Logical address space — a process's view of memory
- Physical address space — that supported by the hardware

But where do addresses come from?

```
MOV r0, 0xfffa620e
```

1. CSCE 451/851
2. Operating Systems Principles
3. Memory Management Basics
4. Steve Goddard
goddard@cse.unl.edu
5. http://www.cse.unl.edu/~goddard/Courses/CSCE451
Address Generation

The compilation pipeline

```plaintext
prog P:
  push...
  inc SP, x
  jmp _foo
end P:

Library Routines:

foo:
  push...
  inc SP, x
  jmp 75

Compilation
Assembly
Linking
Loading
```

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

prog P
: push ...
: inc SP, x
: jmp _foo
: end P

push ...
inc SP, x
jmp 75

1000
1500

0

100

175

1100

1175

Library Routines

Base + Limit registers

Program P's Limit Register

Base Register

Physical Addresses

Logical Addresses

MEMORY EXCEPTION

PAS

Program P's

MAX

y

+
Memory Management Issues

Fragmentation

- Internal fragmentation

- External fragmentation

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
Variable-Sized Partitions
Eliminating Fragmentation

◆ Compaction
   » relocate programs to coalesce holes

◆ Swapping
   » preempt processes & reclaim their memory

10 Executing programs larger than physical memory
Overlays

◆ Program call graph
  prog P
  func A()
  func B()
  A()
  B()
  and A
  D()
  and B
  and P
  calls
  overlays

◆ Memory layout
  Initial:
  Overlay 1:
  Overlay 2: