process = box = a self-contained train of thought

process ≠ program; program might be composed of 1 or more processes
What is inside a process?

**Process struct (in kernel mem):**

- pid (number)
- Saved regs: 
  - eax, ebx, ecx, edx, esi, edi ...
- Virtual address table (pointer)
- File descriptor table (points to open sessions)

**Process struct:**

- PCB (process control block)
- task_struct
How do you execute multiple processes at the same time when num processes > num CPU cores?

Answer: switch really fast, storing regs in process struct

Home slice: duration for which each process runs

Early computers: direct access to physical memory

In practice, we may use pages, not segments (common page size ~ 4KB)
How to deal with file sessions?

Body: one table of sessions

Store sessions per process?
Store sessions external to processes?

Open file table (session table)

Final version: Add per-file table storing info specific to files.

File descriptor tables (session IDs) 

Open file tables (session tables)

Vnode table (file info table)
int fork(); split your process into two

getpid(); returns your pid
getppid(); returns parent pid