Operating Systems

Review

Questions

- What are two functions of an OS?
- What “layer” is above the OS?
- What “layer” is below the OS?

Questions

- What causes OS to change?
  - Or, why aren’t we still running MS-DOS?
- What is a *process?*
- What is a *file?*

True or False

- Unix is a “simple structure” OS
- Micro Kernels are faster than other OSes
- Virtual Machines are faster than other OSes

Questions

- Name 3 operating system structures
- Give one advantage of each
- Give one disadvantage of each

Questions

- How does a shell work? Or … arrange the commands in order:

```
wait()
pid = fork()
exec()
gets()
while(1) {
}
```

Review

- What is a PCB?
- Usually the PCB is in OS memory only. Assume we put the PCB into a processes address space. What problems might this cause?

Review

- List steps that occur during *interrupt*
- True or False:
  - Context switch times happen every 5-10 seconds
  - Most processes have long CPU burst times

Review

- What is (average) waiting time?
- Explain how SJF works
- True or False:
  - FCFS is optimal in terms of avg waiting time
  - Most processes are CPU bound
  - The shorter the time quantum, the better

Questions

- How does Windows NT/2000 avoid process starvation?

Review

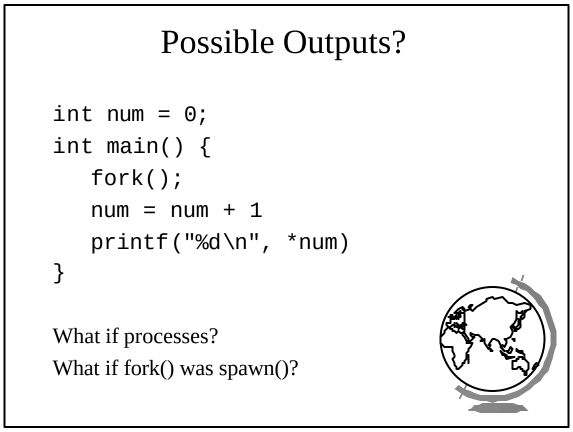
- What is a “race condition”?
- What are 3 properties necessary for a correct “critical region” solution?
- What is Peterson’s Solution?

Possible Outputs?

```
int main() {
   int *num, shm_id

   shm_id = shmget(502)
   num = (int *) shmat(shm_id)
   *num = *num + 1
   printf("%d\n", *num)
}
```

(Assume shm is 0 when first Created)

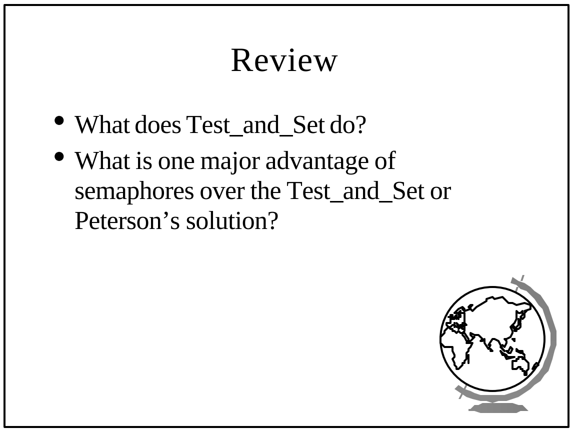


Possible Outputs?

```
int num = 0;
int main() {
   fork();
   num = num + 1
   printf("%d\n", *num)
}
```

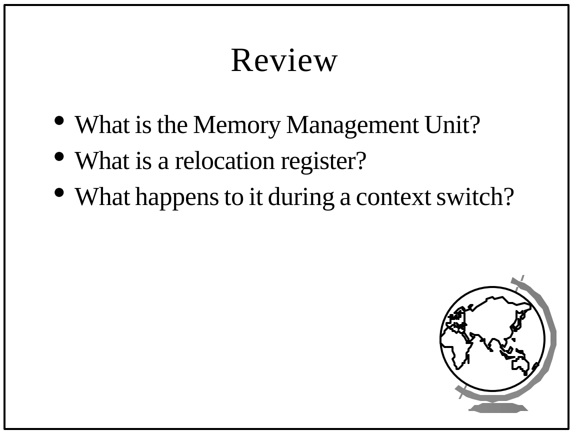
What if processes?

What if fork() was spawn()?



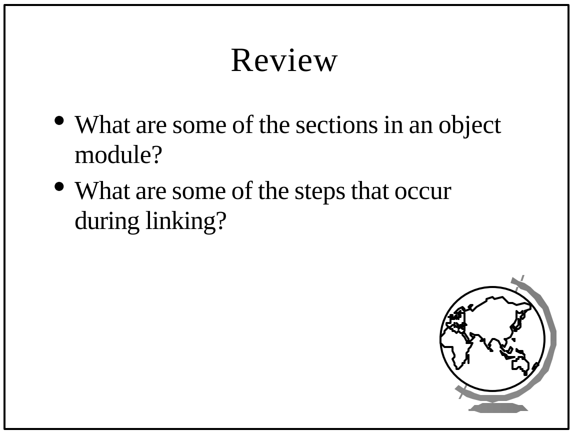
Review

- What does Test_and_Set do?
- What is one major advantage of semaphores over the Test_and_Set or Peterson’s solution?



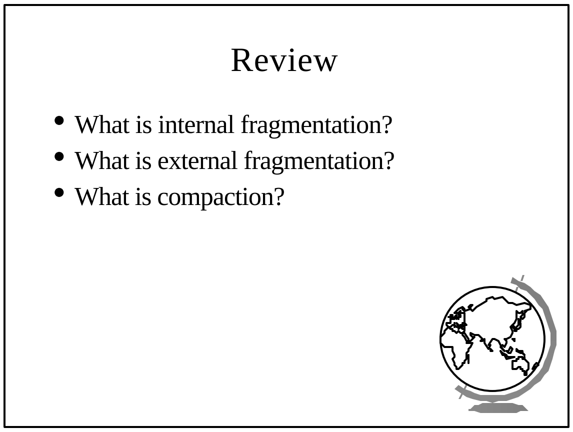
Review

- What is the Memory Management Unit?
- What is a relocation register?
- What happens to it during a context switch?



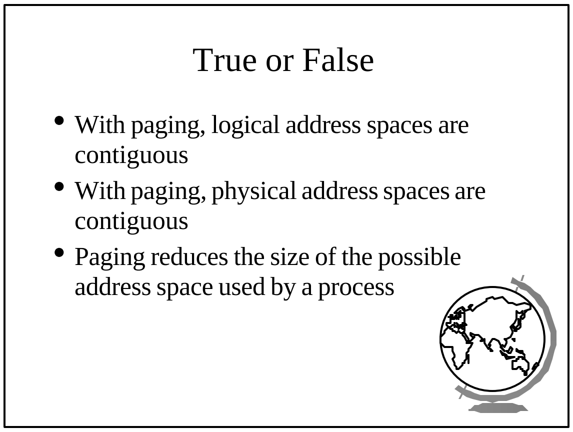
Review

- What are some of the sections in an object module?
- What are some of the steps that occur during linking?



Review

- What is internal fragmentation?
- What is external fragmentation?
- What is compaction?



True or False

- With paging, logical address spaces are contiguous
- With paging, physical address spaces are contiguous
- Paging reduces the size of the possible address space used by a process

Review

- Does paging have fragmentation?
  - – No? Then why not?
  - – Yes? Then what kind?
- What are the overheads associated with paging?

Review

- What is run-time, dynamic linking?

Review

- True or False:
  a) The logical address space cannot be bigger than the physical address space
  b) Processes have big address spaces because they always need them
- Demand paging:
  a) Is unrelated to basic paging
  b) Brings logical pages into physical memory when requested by a process
  c) Increases memory requirements for a system
  d) All of the above
  e) None of the above

Review

- Page faults
  - – What is a page fault?
  - – What does an OS do during a page fault?
- What is a Page Replacement Algorithm?
  - – What is "Belady's Anomaly"?
  - – How does the Optimal algorithm work?
  - – How does Enhanced Second Chance work?
- What is thrashing?
  - – How do we fix it?

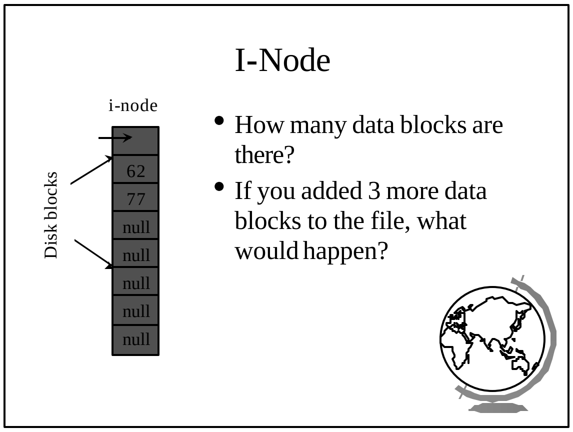
Review

- What is a file descriptor?
  - – What information must it contain?
  - – What information might it contain?

Linked-List with Index

Physical Block	
0	
1	
2	null
3	null
4	7
5	
6	3
7	2

- How many files are there?
- How large are they?
- How many free blocks are there?



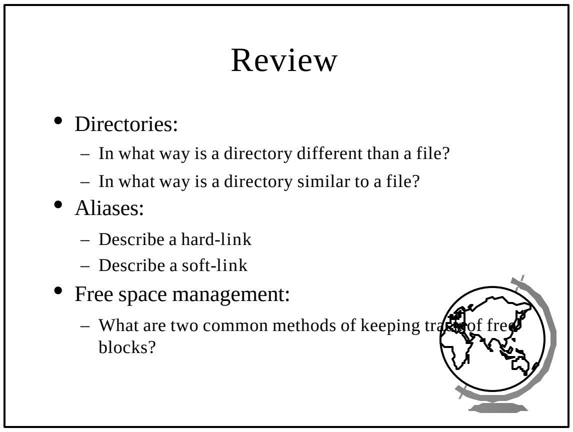
I-Node

i-node

Disk blocks

i-node
62
77
null
null
null
null
null

- How many data blocks are there?
- If you added 3 more data blocks to the file, what would happen?



Review

- Directories:
  - In what way is a directory different than a file?
  - In what way is a directory similar to a file?
- Aliases:
  - Describe a hard-link
  - Describe a soft-link
- Free space management:
  - What are two common methods of keeping track of free blocks?