CPSC 457 Operating Systems

Lecture 9 Page Replacement Algorithms Concurrency

Last Time

Virtual Memory

- Fragmentation
- Paging
- Page Tables
- Translation Look-aside Buffers
- Page Faults
- Locality of Reference

This Time

Virtual Memory

- Thrashing
- Kernel Memory Management Critical Sections
- Memory Management Considerations
- Real World Memory Management

Concurrency

- Page Replacement Algorithms Race Conditions (and other concurrent problems)

 - · Atomic Operations
 - Locks
 - Mutexs
 - Semaphores

What to do when our frames are full?

Page Replacement Algorithms

Optimal

FIFO

Second Chance

Not Recently Used

Least Recently Used

Working Set

CPSC 457 - Tyson Kendon 201

Optimal

CPSC 457 - Tyson Kendon 2016

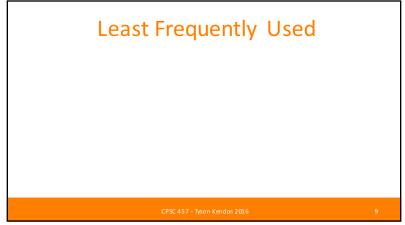
First In, First Out

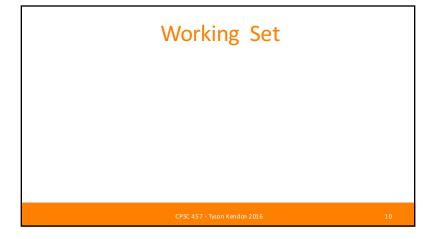
CPSC 457 - Tyson Kendon 201

Second Chance

CPSC 457 - Tyson Kendon 2016









Kernel Memory Allocation

Buddy Allocation

SLAB Allocation

CPSC 457 - Tyson Kendon 201

12

Memory Management Considerations

Local vs Global Allocation

Non-Uniform Memory Access

Page Size Selection

How we code

Page Locking

PSC 457 - Tyson Kendon 2016

13

Real World Memory Allocation

Windows

• Clustering / Working Set

Linux

LRU – With a clock

CPSC 457 - Tyson Kendon 201

14

Concurrency

What are the consequences of running processes at the same time?

What can we do to make sure that processes don't run over each other, while keeping the benefits of parallelism.

CPSC 457 - Tyson Kendon 2016

15

Race Conditions What is the value of counter? A: register1 = counter B: register1 = register1 + 1 P1 C: counter = register1 D: register2 = counter E: register2 = register2 - 1 P2 F: counter = register2

Critical Section

An area of code where processes change something common

- Mutual Exclusion. No two processes may be simultaneously inside their critical regions
- Universality. No assumptions may be made about speeds or numbers of CPUs
- Progress. No process running outside its critical region may block any process
- 4. Bounded Waiting. No process should have to wait forever to enter its critical region

CPSC 457 - Tyson Kendon 2016

17

Atomic Operations

Test_and_Set_Lock

Simultaneously take the value of a variable and set it

CPSC 457 - Tyson Kendon 2016

18

Locks

PSC 457 - Tyson Kendon 2016

19

Mutexes

Lock

Take the lock if available, wait if not

Unlock

Free the lock and wake the waiting processes

PSC 457 - Tyson Kendon 201

Semaphores

Wait

Wait for access to a resource

Signal

Let waiting processes know to proceed

CPSC 457 - Tyson Kendon 2016

2

Next Time

Concurrency

Classical problems

- Producer / Consumer
- Dining Philosophers
- Sleeping Instructor

DeadlockHow locks can break

execution

PSC 457 - Tyson Kendon 20

22