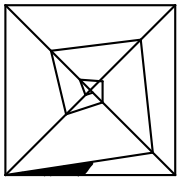
It SNOWED! and was a dangerous drive today :)

for next paper no file system paper about fat.

ntfs, or others can be used.



Shortest job Next - SJN

Job 1	1 min
Job 2	5 sec
Job 3	2 sec

fifo wait time $\frac{(0+60+65)}{3} = \frac{125}{3}$

SJN wait time $\frac{(0+2+7)}{3} = \frac{9}{3}$

Dynamic priority

Shortest remaining time next - SRT

With static priorities starvation is always a concern

Interactive Systems

Long term Scheduling - done by user

Medium term Scheduling.

- assign time quantum to process determine how long it is memory resident. Residents to long can be Skipped out.
- periodic processes run when scheduled not resident otherwise.

Short term scheduling

- Select jobs based on priority
- May be order in ready queue
- Preemption is allowed
- time slicing is used to share CPU

Round - Robin Scheduling

- processes in single queue
- each process is assigned a fixed time quantum
- When CPU is available pick process at front of Queue
- When running process exceeds time quantum it is interrupted and put at back of Queue
- New process becoming ready are queued.

Priority based Round - Robin

- ready queue in a priority queue
- priority is going to change over time.

Feedback Queuer

define

a compute bound process is one that user many time slicer before it performs I/O

I/O Bound: a process that reqyor only one timeSlicr or lers

feedback Queuer give a more favorable service to I/O bound processer by reducing priority of computation bound processes.

Count time slicer to try and solve Starvation for a feedback Queue.

fair share scheduling

you promise a group of users or category of processer a particular level of service

Each group is given a fixed percentage of cpu time.

Lottery Scheduling - UNIT

Suppose we have 1000 time slices / min
each process assigned a number when it enters the ready queue. In a range reflecting the level of service it should receive the category it is in.

Class A - 40% of CPU time
- random number (0 - 0.4)

Class B - 30% of cpu time
- random number (0.4 - 0.7)

Class C - 20% of CPU
- rnd.next (.7 - .7)

class D - 10% of time.
- rnd. next (.9 - 1.0)

11/14/18 Day 32

Last paper modern file system will be photo copied to assess writing of department.

Real time systems

- not typically interactive
  - fuel injection in car
  - traction control
- goals
  - determinism - things are predictable
  - responsiveness
  - user control
  - reliability
  - fail soft operation or fail safe

- hard real time
- soft real time - DAC

- Periodic events - built in schedual
- Aperiodic events

Squeals like a stuffed pig! LOL

Theorem:

if there are m periodic events, event i occurs with period p_i and requires C_i seconds of cpu time to procces then

$$\sum^{M} \frac{C_i}{P_i} \leq 1$$ the system can be schedualed

$i=1$

example:

3 events $e1, \tau 2, \& e3$

periods $P_1 = 100$ msec, $P_2 = 200$ msec, $P_3 = 500$ msec

$C_1 = 50$ msec, $C_2 = 30$ msec, $C_3 = 100$ msec

$$\frac{50}{100} + \frac{30}{200} + \frac{100}{500}$$

$$= \frac{1}{2} + \frac{15}{100} + \frac{1}{5} = \frac{50 + 15 + 20}{100}$$

$$= \frac{85}{100} < 1$$

$\frac{X}{1000}$ if numerator is smaller than $\frac{15}{100}$

Static priority driven preemptive approach

- analyze and assign priority to each possible event
- Schedual based on priority.

Dynamic planning based approach

- feasability determined at run time
- arriving task is accepted only if it feasable.

Dynamic best effort approach

- not going to do feasibility analysis.
- try and meet all deadlines and abort all processes that miss deadline.

Rate monotonic scheduling algorithm - RMS
- a priority assigned in advance to each proces that is proportional to freqyency of triggering events

- Schedule highest priority process first.

Earliest dadline first -
Schedule earlest deadline first.

Least Laxity Schedualing
- time a proces has to spare.

Choose process with smallest laxity to run 1st.

11/16/18 Day 33

- Multi-processor scheduling
  Design issues
  - assignment of processes to processors
  - use of multiprogramming on individual processor
  - dispatching of processes
  - Load balancing can be a problem

block devices
- Stores information in blocks
- each block has its own address
- each block can be, read on or written of each other.

Characters devices - printer, keyboard, mouse, ...
- delivers and accepts a stream of characters
- no block structure
- it is not adaresable
- does not have a seek operation.

Have you heard of travis CI?

I/O units of 2 components - mechanical and electronic
- O.S deals with device controller.
- device controller interacts with mechanical component.

- explained how all disks come with bad sector and how it affects the software.

- Standard interfaces IDE, SATA, SCSI, SAS, etc....
- interface between controller & device is low level.
- what comes out disk is a bit stream.

Preamble	data sector	ECC

Controller's Job

- convert bit stream into a block
- perform any needed corrections to bit stream
- bit stream assembled into a block in the controllers buffer
- check sum corrupted
- Copied to main memory.

How do we communicate with the controller?

Controller communicates with O.S.

I/O address - place in memory where commands are put for controller.

registers -

interrupts - 16 interrupt channels