Event: Brand Montana
Period: Event Brand
Croissant - Bond
- Custom - Impressions
- Event / Trust
- Application Specific
- Small memory feature
- Other
- Part AC + Radio + Power Efficient
- Think ahead Remote volume
- Think ahead Remote volume
- Command open source or fair WSN

5/10 Challenge Key

20.4


Consensus the approach

- Requires complete a priori
  - Conflict - consistency
  - Relativearchy shy: overlapped
    - in
      - complete vs. reply
       - addressing space isolation
       - function
       - unnecessary a overkill
       - large memory shares

- Monolithic Kernel
  - VM
    - I/O
    - Scheduler

- NFS

- I/O
  - Hardware

Translated as,
Ship down memory size of system

Optional of memory size of system

S/v S/send as exception

Dynamic memory alloc

Single linear pool add space

Process MGMT

Only one process (thread)

Direct I/O to main pool

Tim OS Reel

204
- Main (Initialize Scheduler)
  - App Component
  - Frame (Stripe)
  - Task: Component
  - Interface:
  - Event: User/Keypad
  - Single thread stack

- No Kernel/user space discipline
  - Compiled into one executable
  - Application = Scheduler + Graph
  - 205 Overview
Tiny OS

- a component model: app. is broken down into small (reusable) components (modules) and composed into larger abstractions.
- a concurrent-execution model: defines how components interleave their comp. as well as how interrupt & non-interrupt code interact.
- APIs, services, component lib & overall component structure (akin object-oriented paradigm) that simplify writing new apps & services.

Tiny OS Execution Model

- every I/O call is done in split-phase (as opposed to block until completion)
  - a req. returns immediately
  - caller gets a callback when the I/O completes

⇒ one stack suffices
return true if task finished. run task [nextTask].

if (nextTask == NATO.k)
    nextTask = popTask();

return false;

if (NATO.head == nextTask)
    nextTask = NATO.next;

if (NATO.next == nextTask)
    runNextTask();

Scheduler: - Fixed Length - Pro gra.
- runs tasks atomically
- will run later
- any component can place a task after
- tasks are deferred before run call.

Scheduler: - Fixed Length - Pro gra.
- runs tasks atomically
- will run later
- any component can place a task after
- tasks are deferred before run call.

Scheduler: - Fixed Length - Pro gra.
- runs tasks atomically
- will run later
- any component can place a task after
- tasks are deferred before run call.
From chores:

- shn (sc) code that is only reachable
- cshn (ac) code that is reachable from
  - static check for each receiver
  - Restrictions on shared state
    - Data races can occur due to concurrent

Concurrency & Atomically

P1: No nec. long.gap of all flat files

Diagram:

- Command
  - Command
  - Post
  - Preempt Tasks
  - Revert
  - Tasks

2003
Problem 1: Every do and race ends.

is a potential race end.

Claim: Any update to a word after the

Claim: Any update to a word after the

After race can occur because insertion of a race SC

Inverter: SC is atomic or not after SC

Scheidler, Pennock