

# The UNIX Time-Sharing System

## Motivation

- Mostly the the convenience of the authors (programmers).
- Interactive computing vs. batch

## Features

- **General-purpose, multi-user, interactive operating system.**
- **Hierarchical FS; compatible file, device, and inter-process I/O**
- **Asynchronous processes**
- **Inexpensive equipment**

# File System

- **Ordinary files vs. directories**
- **Protection (permissions, SUID bit)**
- **I/O (file descriptors, open(), read(), write())**
- **Implementation (inodes)**
- **How is hierarchy used?**

# Processes

- **Forking, pipes for IPC**
- **Inheritance of attributes from parents**

# Shell

- **What is it?**
- **Standard in/out, pipes, file redirection**
- **Multitasking**
- **Initialization: what happens when you log in?**

# **Flash: An efficient and portable Web server**

## **Motivation**

- **Serve web pages as fast as possible (i.e. maximize utilization of available hardware resources.)**

# Background

- **Basic functions a web server performs**

## **Architectures**

- **MP, MT, SPED, AMPED (how are they same/different?)**
- **AMPED mechanics (how do the helpers work?)**



## **SPED vs. AMPED**

- **What kind of workload is each better at?**
- **How would the relative performance of SPED and AMPED change if all operating systems supported asynchronous disk I/O ?**

## **Other considerations**

- **Other Flash optimizations (basics)**
- **Dynamic content – is there any difference between SPED and AMPED here?**
- **How many AMPED helpers are likely to be useful?**

# **Eliminating Receive Livelock in an Interrupt-driven Kernel**

## **Motivation**

- **Interrupts don't work well (or at all) under high network load.**
- **Example apps: routing, network servers**

## Polling vs. Interrupts

- What is polling? What is an interrupt?
- What is congestive collapse/receive livelock?
  - Why was this not a problem originally?

## **MLFRR (Maximum Loss Free Receive Rate)**

- **What is it?**
- **What do we want to have happen when we reach it?**

## **Solution**

- **What is good about interrupts? Bad?**
- **What is good about polling? Bad?**
- **What is the authors' final solution?**

# **Ethernet: Distributed Packet Switching for Local Computer Networks**

## **Motivation**

- **Cheap, reliable, scalable communication among multiple local area computers**

## Design issues

- **Topology: Decentralization for reliability**
  - Shared medium, decentralized control
  - Advantages/disadvantages of this? How can it fail?
- **Coding**
  - Manchester coding: What is it? What problems does it solve?



## Design issues

- **(some) Mechanisms for decreasing loss**
  - Carrier detection
  - Interference detection (collision detection)
  - Packet error detection
  - Collision consensus enforcement
- **What is exponential backoff, and what problem does it solve?**

# Design issues

- **Growth issues**
  - Signal cover, traffic cover, address cover
  - What is the solution for each?
- **What is exponential backoff, and what problem does it solve?**

## Terminology

- **Packet, Broadcast, Slot, Collision, Carrier Sense, Exponential backoff**
- **Tap, Transceiver, Interface – what's the difference?**
- **Statistical Arbitration**

## Questions

- **What is the packet format? How is addressing done? What is the CRC for?**
- **What other network topologies are there? How can they fail?**
- **Does Ethernet guarantee that packets will get delivered?**
- **Why is there a minimum packet size? How do you calculate it? (was NOT in the paper, but good to understand).**