

# 20<sup>th</sup> ACM Symposium on Operating Systems Principles

---

SOSP '05

Brighton, Reino Unido  
23-26 de outubro de 2005  
<http://www.sosp-20.com>

Daniel de Angelis Cordeiro

# O que é o SOSP?

---

O SOSP é uma conferência bienal que reúne profissionais da área acadêmica e privada interessados em tecnologias relacionadas à sistemas operacionais.

É promovida pelo *Special Interest Group on Operating Systems* (SIGOPS) da ACM e sua primeira edição foi em 1967.

# Estrutura da Conferência

---

Os trabalhos foram apresentados ao longo de 7 sessões:

- *Integrity and Isolation;*
- *Distributed Systems;*
- *History and Context;*
- *Containment;*
- *Filesystems;*
- *Bugs;*
- *Optimization.*

# Estrutura da Conferência

---

Além disso, as seguintes atividades ocorreram:

- Sessão de Posters;
- *Bird of a Feather*;
- Painel: *Peer-to-Peer: Still Useless?*;
- *Work-in-progress*;
- Entrega dos prêmios Mark Weiser e SOSP Hall of Fame.

# SOSP Hall of Fame Award

---

Premia os trabalhos apresentados na última década que mais influenciaram a área.

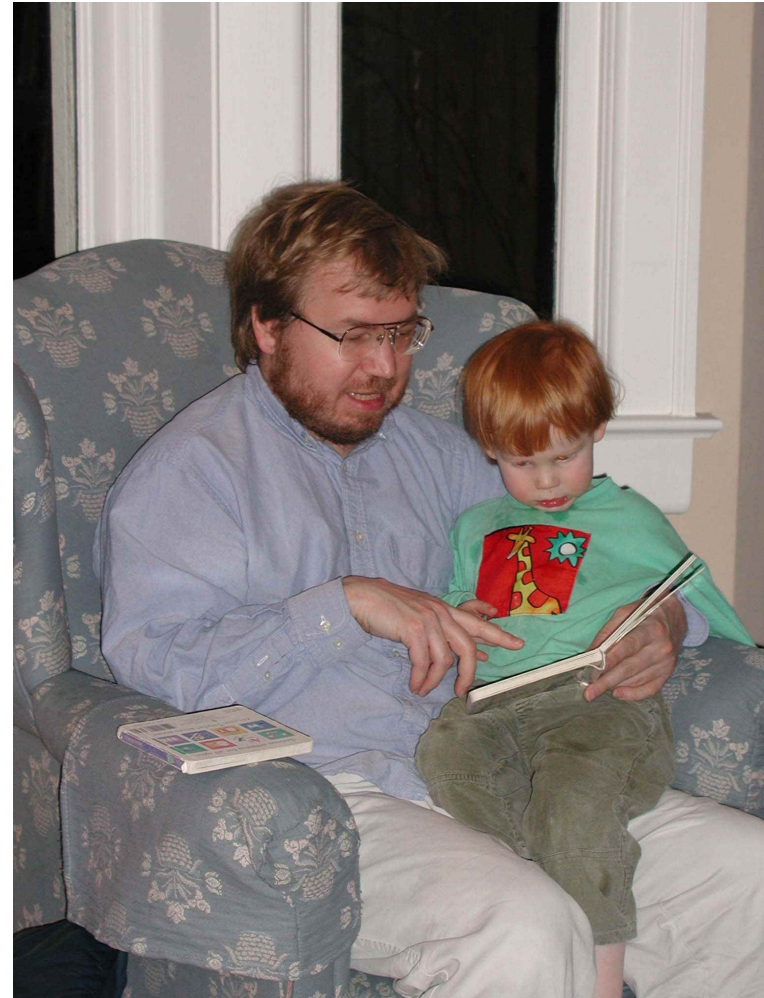
Os ganhadores deste ano foram:

- Dijkstra, "*The structure of the THE multiprogramming system*" (SOSP-1)
- Denning, "*The working set model for program behavior*" (SOSP-1)
- Ritchie and Thompson, "*The UNIX time-sharing system*" (SOSP-4)
- Lampson, "*Hints for computer system design*" (SOSP-9)

# Mark Weiser Award

Thomas Anderson, da Universidade de Washington, foi o ganhador deste ano.

Dentre seus trabalhos na área de sistemas, podemos destacar o SO Nachos, o xFS, o sistema one.world de computação ubíqua e seu trabalho com Scheduler Activations.



# Alguns trabalhos

---

Apresentarei brevemente os seguintes trabalhos apresentados no SOSISP deste ano:

- Vigilante: End-to-End Containment of Internet Worms;
- Speculative Execution in a Distributed File System;
- RaceTrack: Efficient Detection of Data Race Conditions via Adaptive Tracking.

# Vigilante

---

## **Vigilante: End-to-End Containment of Internet Worms**

Manuel Costa (Microsoft Research), Jon Crowcroft (Cambridge University), Miguel Castro, Antony Rowstron, Lidong Zhou, Lintao Zhang and Paul Barham (Microsoft Research)

Vigilante é um sistema capaz de detectar novos *worms* e impedir que se espalhem pela rede. Ao contrário de outros trabalhos na área, o bloqueio não é feito com filtros na camada de rede.



# Vigilante

---

Atualmente usa dois métodos para detecção de *worms*:

- *non-executable pages*:
  - usa proteção contra execução nas páginas de memória que contém a pilha e o *heap*, o que detecta e previne ataques do tipo *code-injection*
- *dynamic data-flow analysis*:
  - monitora fluxos de dados provenientes de certos tipos de entradas perigosas, como a rede. Esses dados são marcados como “sujos” e seu uso é restrito. Ex.: um dado sujo não pode ser carregado no *program counter*.

# Vigilante

---

- Caso um *worm* seja detectado, um *self-certifying alert* (SCA) é gerado. Um SCA é uma prova de que a vulnerabilidade existe e pode ser verificado por qualquer máquina.
- A verificação é feita através de uma execução instrumentada do serviço afetado pelo *worm* em uma VMM (*virtual machine monitor*).
- O artigo discute como propagar as SCAs, qual o custo da verificação de um SCA e como criar filtros para conter os *worms* utilizando as informações do SCA verificado.

# Speculator

---

## **Speculative Execution in a Distributed File System**

Edmund B. Nightingale, Peter Chen, Jason Flinn  
(University of Michigan)

Speculator provê ao núcleo do Linux a capacidade de realizar execuções especulativas. Permite que vários processos compartilhem estado especulativo através de um mecanismo de dependência e garante a corretude da execução ao evitar que o processo externalize o estado especulado.

# Speculator

---

- O desenvolvimento do *Speculator* foi motivado pela vontade de criar uma versão especulativa do NFSv3.
- Os autores acreditam que sistemas de arquivos distribuídos são adequados para execução especulativa pois:
  - os resultados da execução especulativa são muito previsíveis;
  - *checkpointing* em geral é mais rápido do que I/O remoto;
  - computadores modernos possuem recursos ociosos (processador e memória).

# Speculator

- Antes de cada operação especulativa, é realizado um *checkpoint* do processo (basicamente um *fork* onde o novo processo não vai para a fila de processos prontos para serem executados).
- Caso o resultado da especulação se mostre incorreto, o processo é terminado e o processo resultante do *fork* (*checkpoint*) é inserido na fila de processos prontos e populado com as informações do processo original (PID, *thread group identifier*, sinais pendentes, etc.).
- O *Speculator* tornou o NFS 2 vezes mais rápido em LANs e 14 vezes mais rápido em uma rede com latência de 30ms.

# RaceTrack

---

## **RaceTrack: Efficient Detection of Data Race Conditions via Adaptive Tracking.**

Yuan Yu, Thomas Rodeheffer (Microsoft Research), Wei Chen (UC Berkeley)

RaceTrack é uma ferramenta para detecção de condições de corrida em ambientes *multi-threaded* que monitora a execução de um programa e emite alertas sempre que um padrão de acesso suspeito for detectado.

# RaceTrack

- O RaceTrack estende a *Common Language Runtime* da *Microsoft* e pode monitorar qualquer aplicação sem que modificações sejam necessárias.
- Toda variável tem seu *lockset* e seu *threadset* monitorados.
- Para que seja possível monitorar aplicações muito grandes, o RaceTrack ajusta dinamicamente a granularidade das detecções e a técnica de monitoração utilizada. Por exemplo, inicialmente apenas instâncias são monitoradas. Na ocorrência de uma condição de corrida cada variável da instância passa a ser monitorada separadamente.

# RaceTrack

---

Os *bugs* detectados pelo RaceTrack na biblioteca padrão do CLR foram classificados pelos autores e alguns exemplos são listados abaixo:

- alarmes falsos: *fork/join*, protocolo do usuário (8)
- *write* protegidos por *lock*, *read* desprotegidos (4)
- *double-checked locking* (4)
- código com funcionalidade de *cache* (2)
- *unlocked lazy initialization* (7)
- *bugs* muito complicados para entender (8)
- *bugs* sérios (8)



# Keynote Speech

---

- **Andrew S. Tanenbaum** alertou para o problema do crescimento descontrolado dos sistemas operacionais. Para ele, tanto o Windows como o Linux sofrem desse problema;
- Será que os usuários finais percebem os efeitos da evolução do *hardware* na mesma proporção?
- Sistemas operacionais deveriam ser mais flexíveis: deveríamos poder remover os componentes do sistema operacional que são desnecessários;
- Também deveriam ser mais simples: deveriam conter apenas o essencial. O resto deveria estar em espaço de usuário.

# Minix 3

---

Toda essa conversa foi para apresentar a nova versão do Minix:

- *kernel* pequeno (3.800 linhas de código);
- baixa latência de interrupção ( $\sim 10$  microsec.);
- todo *device driver* é um processo distinto, que roda em nível de usuário (exceto o relógio);
- API simples, com apenas 35 chamadas de sistema;

Além de ser um *software* educacional, o Minix almeja ser utilizado em sistemas embarcados e em sistemas de alta confiabilidade.

# Vá ao SOSp '07

---

- Incentive seu aluno a ir ao próximo SOSp!
- Cerca de 65 alunos receberam o *student scholarship* oferecido pelo SOSp neste ano.
- Não há limite fixo para o valor do auxílio. A viagem e os custos com a conferência podem ser inteiramente cobertos com o dinheiro do *scholarship*.
- Os patrocinadores da conferência são **muito** generosos!

# Aos alunos:

---

- O SOSP é um lugar muito legal para discutir assuntos interessantes com gente muito boa;
- Conferências desse tipo também promovem oportunidades muito interessantes de intercâmbio cultural. Tive a oportunidade de conversar com gente da França, Polônia, Japão, Taiwan, Portugal, Estados Unidos, etc.;
- É também uma oportunidade muito boa para ter outras opiniões sobre o seu trabalho. Infelizmente, o meu anda tão enrolado que não foi o caso, mas eu pude dar palpite no trabalho alheio. :)

# Como se isso não bastasse...

... a escolha da cidade para a realização do evento é feita à dedo.



# E trata-se de férias pagas. :)



[visita a um autêntico pub londrino]

# Perguntas?

---