Reincarnation of Dead Device Drivers

Jorrit N. Herder, Herbert Bos, Ben Gras, Philip Homburg, Andrew S. Tanenbaum
{jnherder,herbertb,beng,philip,ast}@cs.vu.nl
Vrije Universiteit, Amsterdam, The Netherlands

Research Summary

**Problem:** Since bugs in software are a fact of life, failures in device drivers and other critical OS components are inevitable.

**Goal:** We want to build an OS that is resilient to such failures so it can continue to work under adverse conditions.

**Results:** We have built a new OS that is:

1. fully compartmentalized by running all components as unprivileged user-mode processes protected by the MMU, and
2. constantly monitored by the special **Reincarnation Server** that can replace malfunctioning parts with a fresh copy.

The Operating System is Called MINIX 3
(note that this poster presents work-in-progress)

Architecture of a Resilient Operating System

![Architecture Diagram]

- **Reincarnation Server** Monitors the System and Automatically Replaces Dead Drivers

---

**Ethernet Driver Recovery**

- User
- wget
- Rp
- lynx

- ressend packet
- Realtek Driver
- IntelPro Driver
- 3Com Driver

**Full Recovery at Network Server**
(transparent to application)

---

**Block Driver Recovery**

- User
- vlm
- dd

- redo I/O
- SATA Driver
- RAM Driver
- Floppy Driver

**Full Recovery at File Server**
(transparent to application)

---

**Character Driver Recovery**

- User
- retry
- mp3
- cdr
- redo job
- Audio Driver
- SCSI Driver
- Printer Driver

**Partial Recovery at Application**
(transparent to user)

---

Supported by the Netherlands Organization for Scientific Research (NWO) under grant 612-060-420.