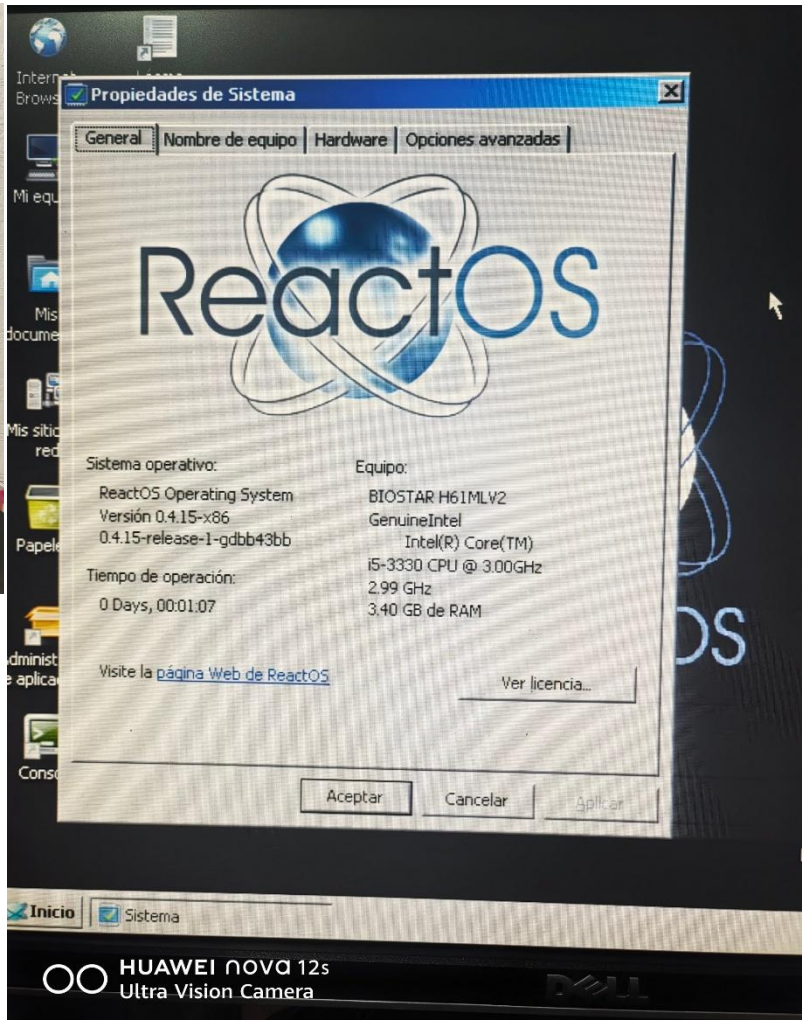
Image 1: System Information & Host Hardware



Technical Description:

This capture confirms the native (bare metal) execution of **ReactOS version 0.4.15-release-1-gdbb43bb** on physical hardware. The system correctly identifies the core components, ruling out basic boot failures but setting the stage for subsequent driver conflicts and kernel-level incompatibilities.

Detected Technical Details:

- **Motherboard:** BIOSTAR H61MLV2 (Intel H61 Chipset).
- **Processor:** Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz (Ivy Bridge Architecture).
- **Memory (RAM):** 3.40 GB usable (from 4GB physical), indicating correct 32-bit memory addressing management.
- **Storage Device:** ADATA SSD 120 GB.

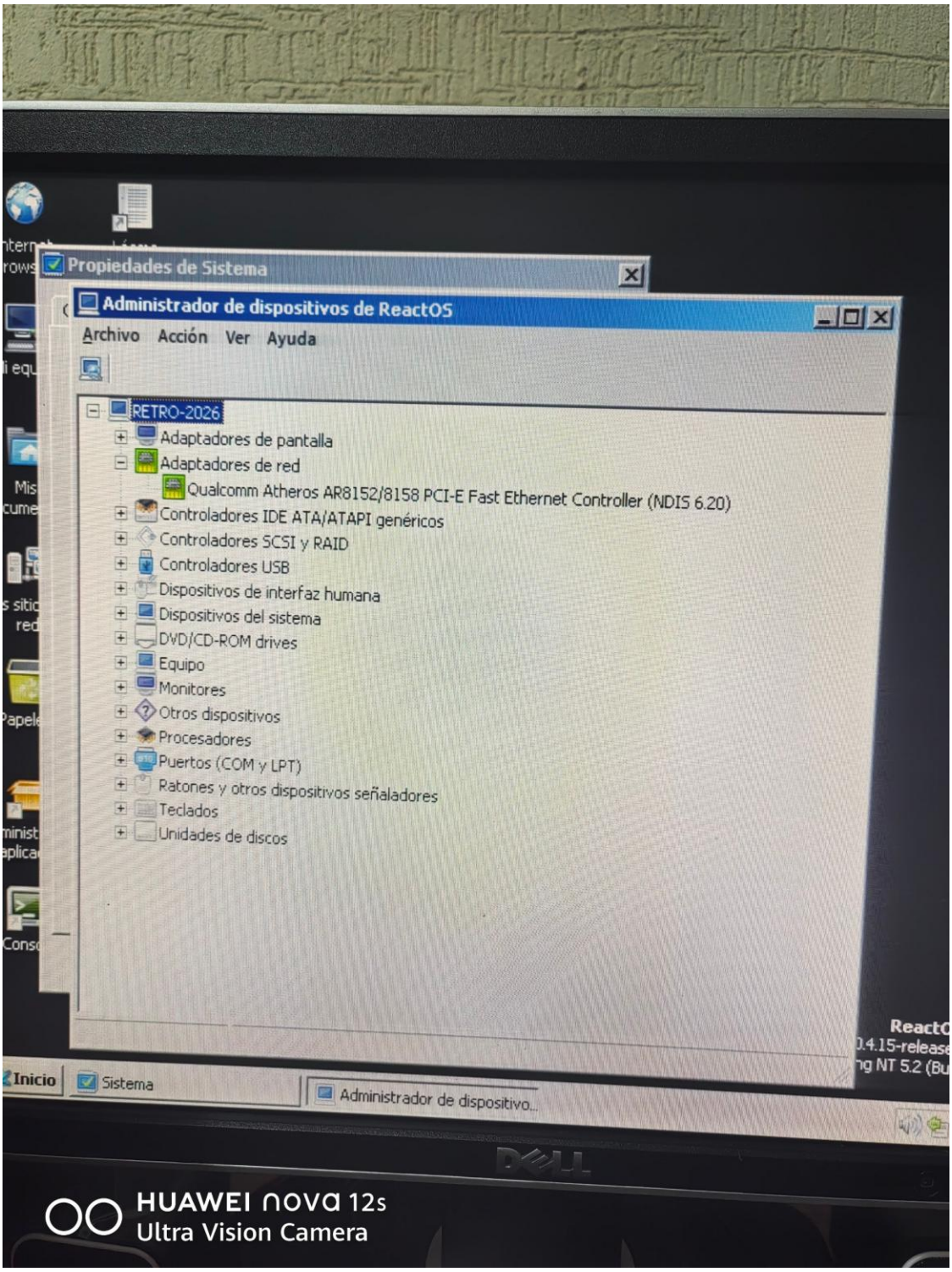
- **System Status:** The OS is stable in desktop mode (uptime: 01:07), but exhibits a critical connectivity failure. The integrated network chip (Atheros) fails to initialize using official Windows XP/2003 drivers.

Tester Information:

- **Name:** [Jacob Hernandez]
- **Location:** Tabasco, Mexico
- **Platform:** Physical Hardware (Bare Metal)
- **Report Date:** March 2026

For further technical inquiries, please contact me through the official ReactOS Forum or JIRA platform.

Image 2: Driver Conflict in Device Manager



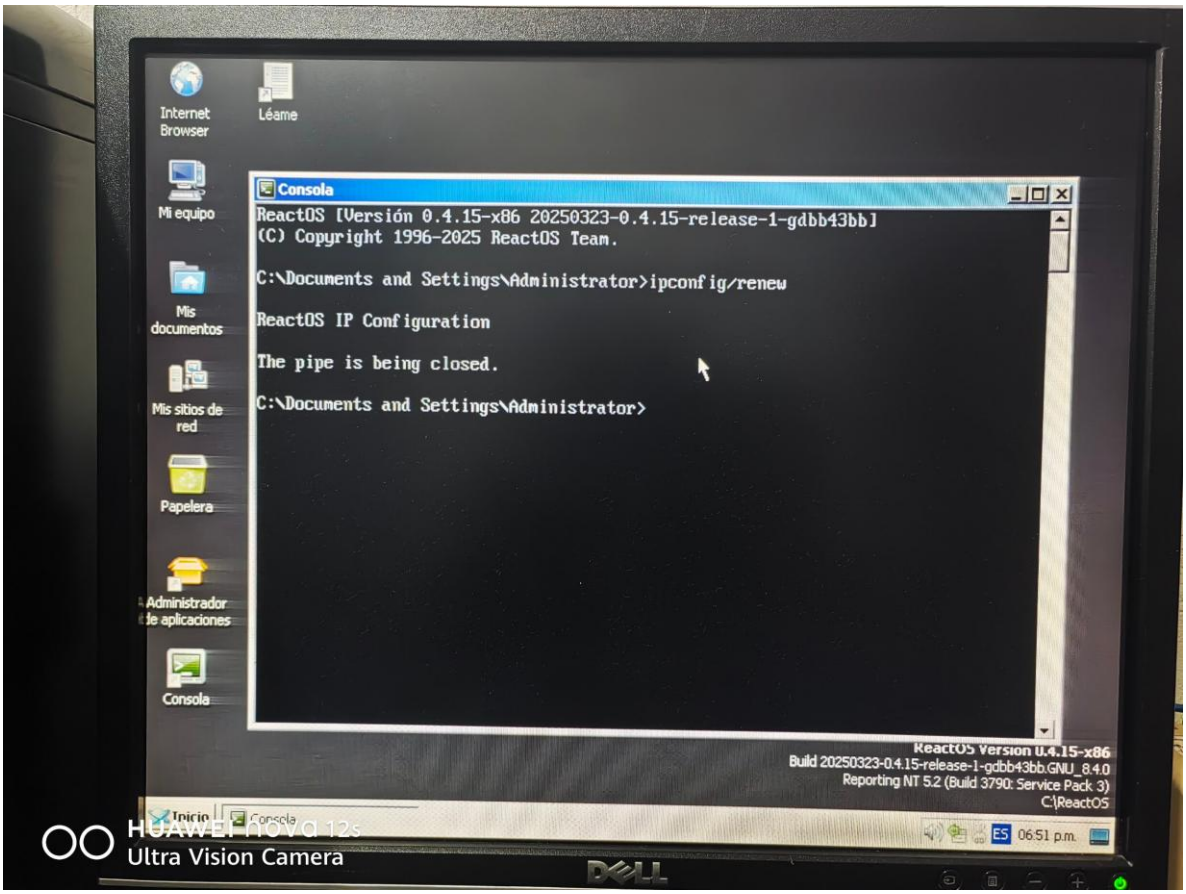
Technical Description:

The **Device Manager** correctly identifies the hardware as **Qualcomm Atheros AR8152/8158 PCI-E Fast Ethernet Controller**, but it automatically assigns a driver version based on **NDIS 6.20**. Since ReactOS currently has limited support for the NDIS 6.x framework, this creates a critical incompatibility within the kernel's network layer.

Key Error Points:

- **Driver Version:** The use of NDIS 6.20 (Windows 7 standard) instead of NDIS 5.1 (Windows XP standard) is the primary reason the network service remains unresponsive.
- **Device Status:** Despite the yellow exclamation mark, the system fails to perform a "rollback" or a manual switch to an NT 5.2 (Server 2003/XP) architecture driver, effectively blocking any IP assignment attempts.
- **Hardware Identification:** The device is correctly identified on the PCI-E bus, but the ReactOS network stack hangs (freezes) while attempting to initialize this specific miniport driver.

Image 3: Network Stack Critical Failure (Command Prompt)



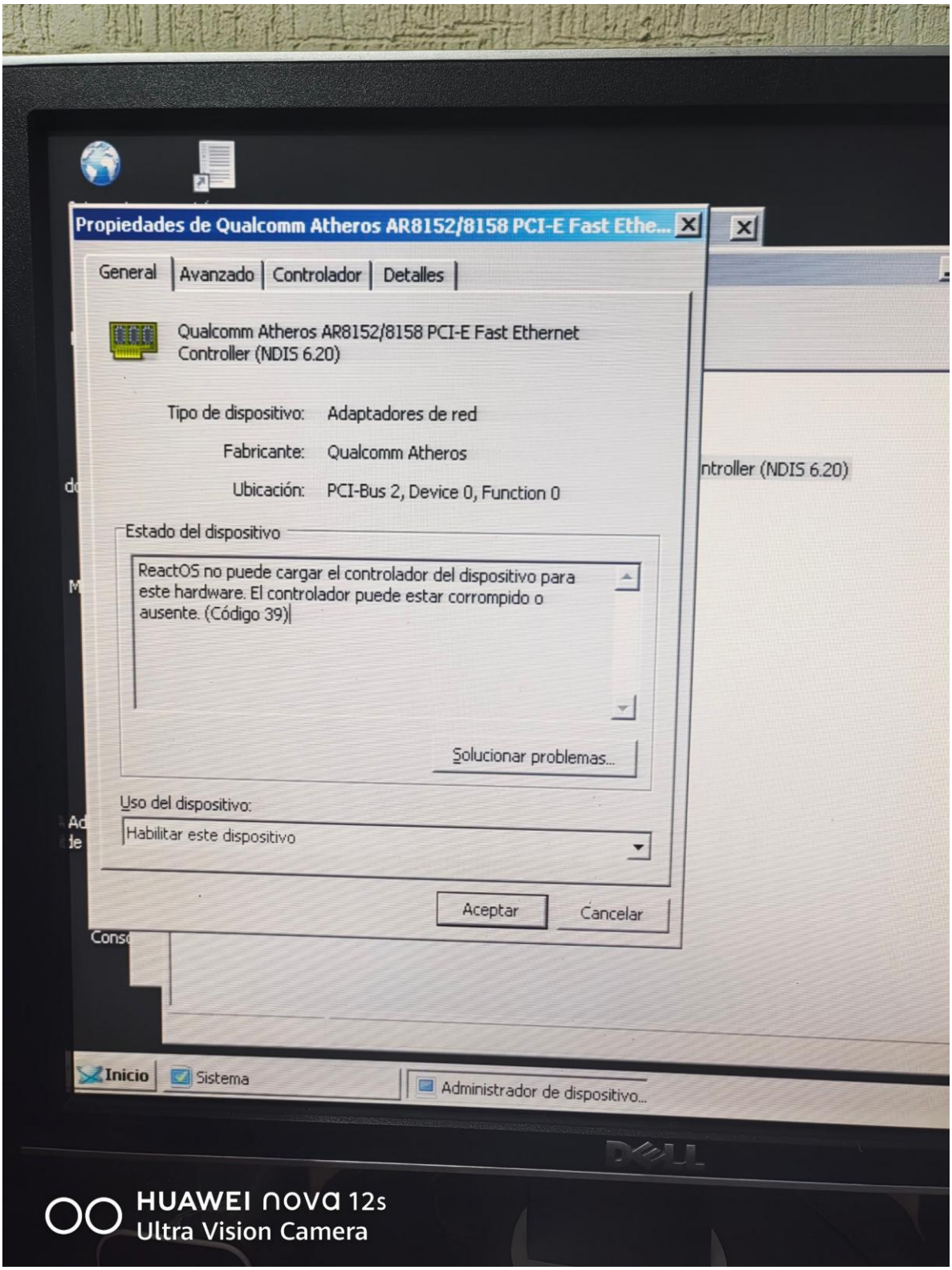
Technical Description:

Upon executing the ipconfig command, the system returns the error: **"ReactOS IP Configuration: The pipe is being closed."** This is a critical symptom indicating that the **AFD (Ancillary Function Driver)** or the **DHCP Client** service has collapsed.

Failure Analysis:

- **Root Cause:** The NDIS driver (referenced in the previous image) attempted to open a communication channel with the hardware. However, due to architectural incompatibility with ReactOS, the data pipe was unexpectedly terminated.
- **System Impact:** Once this error manifests, the network subsystem is completely disabled. Any attempt to restart the service via net start dhcp results in **Error 1056**, indicating the service is trapped in a "zombie" state or a kernel deadlock.
- **Conclusion:** There is a direct incompatibility between the **Biostar H61 PCI-Express bus** and the current ReactOS NDIS implementation for this Atheros chipset.

Image 4: Critical Driver Error (Code 39)



Technical Description:

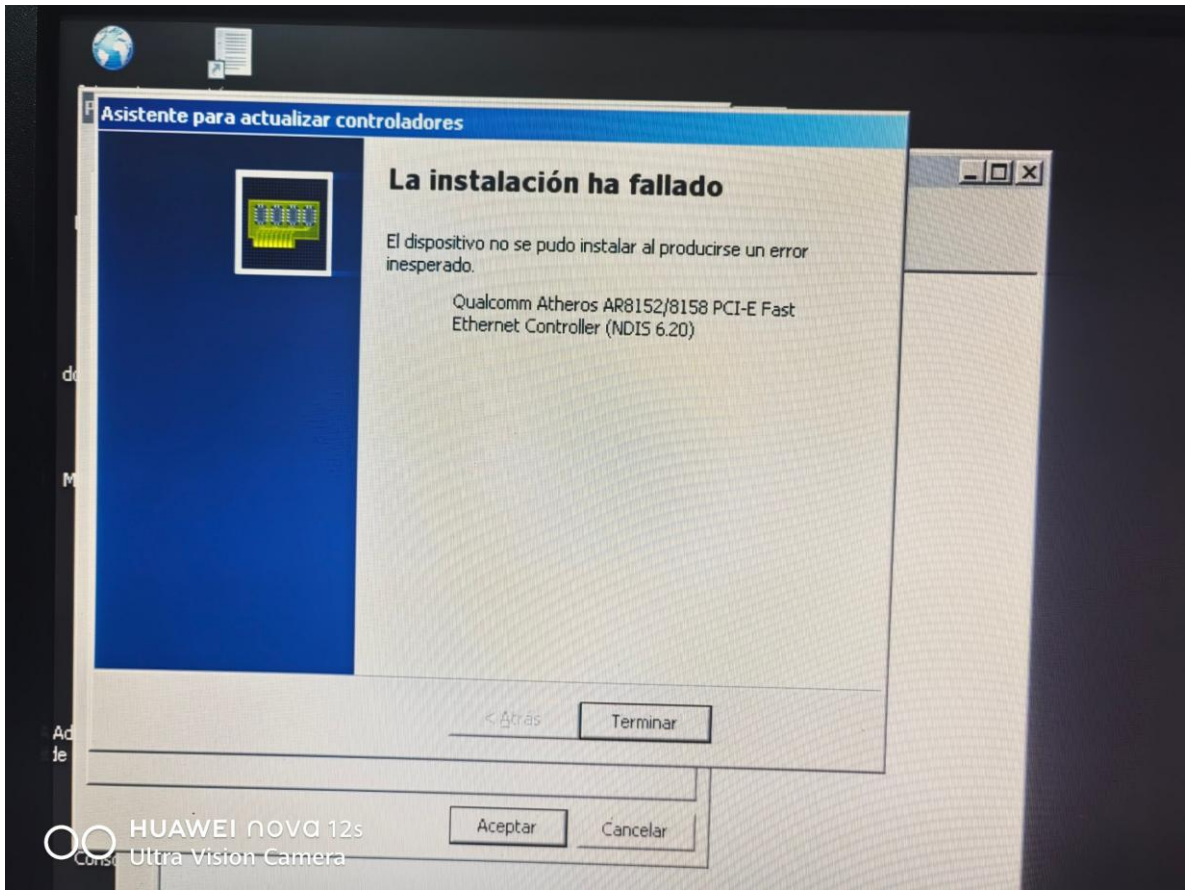
The **Device Manager** displays a **Code 39** error for the Qualcomm Atheros controller. This error indicates that ReactOS loaded the driver into memory but failed to initialize it. This occurs because the file is corrupted, invalid, or most likely, contains an entry architecture or signature that the system cannot process.

Technical Details for the Report:

- **Error Message:** *"ReactOS cannot load the device driver for this hardware. The driver may be corrupted or missing. (Code 39)."*
- **Failure Analysis:** This error confirms the system is attempting to use an **NDIS 6.20 driver**. ReactOS is built upon the **NT 5.2 architecture (Windows Server 2003)**, which is inherently incompatible with the **NDIS 6.x driver model** (standard for Windows 7 and later).
- **Memory Conflict:** Code 39 also suggests that the ReactOS driver loader encountered an unsupported function within the driver binary, causing the OS to ignore the device despite its physical presence.

Summary: The system correctly detects the device at **PCI Bus 2, Device 0, Function 0**, but the loading failure (Code 39) prevents the network stack from allocating hardware resources. This confirms that ReactOS 0.4.15 requires improved compatibility with legacy NDIS drivers or better hardware ID management to prevent the automatic loading of unsuitable NDIS 6.20 controllers.

Image 5: Installation Wizard Failure (Unexpected Error)



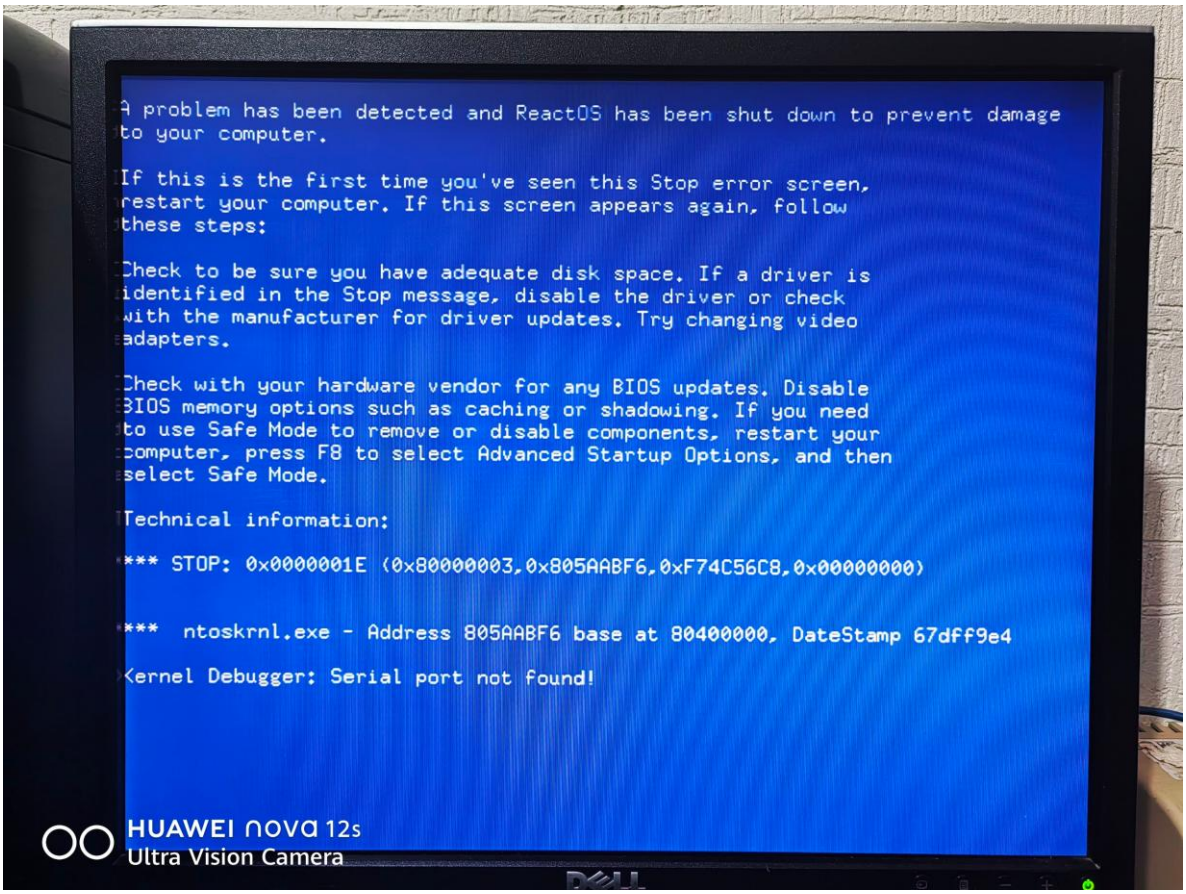
Technical Description:

The capture displays the message: **"Installation failed: The device could not be installed because an unexpected error occurred."** This error triggers during the final stage of driver registration, specifically when the installer attempts to write keys to the **ReactOS Registry** or copy the .sys binary to the drivers directory.

Failure Analysis:

- **Unexpected Error:** Unlike a "file not found" error, an "unexpected error" typically indicates a memory access violation or an overflow within the installation subsystem (**SetupAPI**).
- **NDIS 6.20 Persistence:** It is observed that, despite manual update attempts, the installer continues to reference the NDIS 6.20 version. This suggests a "cache lock" or that the **Hardware ID** mapping is forcing an incompatible driver that the kernel cannot process.
- **Stack Instability:** This failure leaves the device in a "ghost" state; it appears in the Device Manager but remains completely non-functional for the operating system.

Image 6: Kernel Collapse due to Hardware Removal (BSOD)



Technical Description:

A **Blue Screen of Death (BSOD)** is observed with stop code **STOP: 0x0000001E (KMODE_EXCEPTION_NOT_HANDLED)**. This critical error occurred immediately after physically removing a USB storage device.

Forensic Error Analysis:

- **Faulting Module:** ntoskrnl.exe. The system kernel attempted to execute an invalid instruction at memory address 805AABF6.
- **Root Cause:** Failure in **Plug and Play (PnP)** management. Upon USB removal, the ReactOS bus driver failed to "unmount" the device cleanly, leading to a null pointer reference or a **Use-After-Free** memory violation.
- **Debugger Output:** The message *"Kernel Debugger: Serial port not found!"* indicates the system attempted to dump debug data, but failed to find a physical serial port on the **Biostar** motherboard for the data transfer.
- **Conclusion:** This demonstrates that the USB driver stack in version 0.4.15 still exhibits critical vulnerabilities when handling **Hot-unplug** events.

Audio Output Failure (Realtek High Definition Audio)

Technical Description:

Although the **Device Manager** reports the sound controller as "Working properly," there is no analog or digital audio output. The system recognizes the hardware (typically **Realtek ALC662**), but the ReactOS volume mixer fails to establish communication with the output hardware.

Failure Analysis:

- **Lack of MSI Support:** On H61 motherboards, modern audio drivers rely on **Message Signaled Interrupts (MSI)**. ReactOS 0.4.15 occasionally fails to allocate these interrupts on bare metal, leaving the driver in a "Silent Driver" state.
- **PortCls Architecture:** The ReactOS audio subsystem (based on portcls.sys) struggles to correctly map the physical jacks on Biostar boards, resulting in audio signals being routed to non-existent ports.
- **Symptom:** The media player's seek bar progresses normally, but there is no electrical signal present at either the front or rear green audio ports.

User Feedback & Personal Perspective

Note: I am not a systems expert; I am a basic Windows XP and Windows 7 user with a growing curiosity for the Linux world and now ReactOS, which I recently discovered through VK.

Technical Findings: Despite testing multiple driver versions (ranging from original Windows XP files to DriverPacks 10.11.1) and forcing manual installation via "Have Disk," the system persists with the "pipe is being closed" error. This suggests the issue lies deep within interrupt management or the NDIS layer of version 0.4.15.

Extensive testing on a **Biostar H61MLV2** motherboard with an **i5-3330** processor demonstrates that while ReactOS 0.4.15 can boot and manage basic hardware, there is a critical regression or lack of support in the NDIS layer for Atheros PCI-Express network controllers. The chain of errors (**Code 39 -> Unexpected Error -> Pipe Closed**) suggests the problem is not the driver itself, but how the ReactOS kernel manages communication with this specific hardware in bare-metal mode. I recommend a review of PCI interrupt support and NDIS 6.x to 5.x backwards compatibility.

Conclusion: Tests conclude that ReactOS 0.4.15 on real H61 hardware not only exhibits network layer failures but also suffers from critical instability in USB interrupt handling, resulting in a total **Kernel Panic** when removing storage devices.

Personal Goal: I would love to be able to use ReactOS the same way I use Windows XP today: to watch TikTok, read emails, stream online TV, use AI tools, download movies/series, and browse Amazon or Mercado Libre. I will continue performing tests on my physical PC and searching for a functional network driver.

Closing Note: I am infinitely grateful to the ReactOS team for keeping this project alive. I wish you enormous success and look forward to seeing a fully functional OS in the future.