



Newsletter of the North Orange County Computer Club

range Bytes

Volume 50 No 11

November 2026

\$1.50

NOCCC meetings for Sunday Nov 1, 2026

MAIN MEETING FOR Nov 1st

Jim Sanders will be demonstrating several
Utility/productivity programs

NOTE. A few lapsed members are receiving the Bytes as an encouragement to re-up and attend.

Special Interest Groups (SIGs) & Main Meeting Schedule

9:00 AM – 10:30 AM

Beginners Digital Photography **Science 131**

Q and A about Digital Photography, need new leader

Linux for Desktop Users..... **Science 129**

Beginners' Questions about Linux

10:30 AM – 12:00 PM Noon

3D Printing **Science 127**

Questions and Answers about 3D printing if requested

Advanced Digital Photography... **Science 131**

Questions and Answers about Digital Photography

Linux Administration **Science 129**

More topics about the Linux operating system

Mobile Computing **Science 109**

We discuss smart phones, tablets, laptops, operating systems and computer related news. **Waiting for a new leader.**

12:00 PM Noon – 1:00 PM

3D Printing..... **Science 127**

Questions and Answers about 3D printing if requested.

PIG SIG **Irvine Courtyard**

Bring your lunch. Consume it in the open-air benches in front of the Irvine Hall or join the group that goes to the student cafeteria. Talk about your computer(s) and life experiences.

1:00 – 3:00 PM Main Meeting

What is going to happen during the main meeting in November? Jim Sanders will be demonstrating some useful free programs. There are a surprisingly large number of FREE Windows programs. (Yes, President Strain, I know Linux stuff is free, free, free.) And a number of them are in the same sub category of Utilities or Productivity software..

There are 3-4 that I will spend some time on and several that I will list. Among my favorites, CPU-Z, CrystalDiskInfo, Everything, HWINFO.

BOD.....3-4PM..... Science 129

Verify your membership renewal information by checking your address label on the last page. If it is not right, let the treasurer know.

Mark your calendars for these meeting dates

2026: Nov. 1, 1Dec. 6,

2027: Jan. 10, Feb. 7, Mar. 7, Apr. 4, May 2, Jun. 6, maybe?

Coffee, cookies and donuts are available during the day in room 129 .

“Friends Helping Friends” since April 1976

Table of Contents

Main Meeting ___ 1

Special Interest Groups ___ 1

Again, verify your membership renewal month information by checking your address label on the last page

Mark your calendars for these meeting dates ___ 1

Contact information and email forwarding addresses ___ 2

Editors Corner ___ 2

The UEFI Boot Nightmare ___ 2

Directions and map ___ 4

Special email addresses

Jim Sanders is: editor@noccc.org

membership@noccc.org

Our Website

WWW.NOCCC.ORG

Reminder: Membership expiration dates are based on the date that you joined the club.

Example, you joined or re-upped your membership in the club in December of 2025.

That means that in December 2026 you should pay your membership dues. In the address label area of the Orange Bytes is your join month/expiration month.

A thought to consider; The only reason the club has been able to continue functioning financially after Covid is a generous donation in the will of a former member. Could you include a small donation in yours?

Reprint Policy

Permission to reprint articles is granted to non-profit groups and software/hardware vendors whose products we review, provided North Orange County Computer Club is credited as the source and the original author is given full credit. We request that any non-profit that reprints one of our members' articles send a PDF copy of your newsletter to the NOCCC Orange Bytes editor: editor@noccc.org.

Board of Directors

Contact information and email forwarding addresses

President Robert Strain

president@noccc.org (cell 714.222.2140)

Vice President (acting) Jim Sanders

vicepresident@noccc.org (714.544.3589)

Secretary position is open

secretary@noccc.org

Treasurer Dr. Don Armstrong

treasurer@noccc.org (714-343-1121)

Webmaster Jim Sanders

webmaster@noccc.org (home 714.544.3589)

Director Terry Dickson

terry@noccc.org (home 714.899.9913)

Director Dennis Martin

dennis@noccc.org (home 951.926.3065)

Director Richard Miller

richard@noccc.org (cell 909.955.2140)

Editors Corner

The UEFI Boot Nightmare: Why Accessing Your Motherboard Now Requires a Password

The illusion of total ownership over personal computing hardware has slowly eroded over the past two decades. There was a time when turning on a computer meant engaging directly with the physical machine, a relationship mediated by clear, unyielding firmware commands built into the motherboard. Today, that relationship is heavily policed by layers of operating system intervention, cryptographic handshakes, and defensive software architectures. What used to be a seamless physical interaction has transformed into a bureaucratic digital gauntlet. This shift is perfectly illustrated by the evolution of accessing a computer's setup utility—moving from the classic, unconditional use of the Delete and F2 keys to the highly regulated, Windows-dependent protocols of modern Unified Extensible Firmware Interface (UEFI) ecosystems.

In the golden era of personal computers, accessing the BIOS was a democratic, hardware-level certainty. The moment the power switch was flipped, the motherboard initialized a Power-On Self-Test (POST). During this brief window, before any operating system could even begin loading from a storage

drive, the system actively listened for a physical interrupt signal from the keyboard. Mashing the **Delete key or F2 key** was a universal tech-support ritual. It did not matter if your operating system was completely corrupted, if your hard drive had suffered a catastrophic mechanical failure, or if you were booting a blank home-built rig for the first time. The firmware belonged to the user, and the keypress was an absolute override that granted immediate access to low-level hardware configurations, boot order priorities, and voltage controls.

As cyber threats evolved, the open architecture that allowed users such seamless control also left systems vulnerable to low-level exploits, such as rootkits and boot-sector malware that could compromise a machine before an operating system's security features could initialize. The computing industry responded by replacing the legacy BIOS with UEFI and introducing **Secure Boot**. Secure Boot established a chain of trust, ensuring that only firmware components signed with cryptographic keys recognized by the hardware manufacturer could execute during the boot sequence.

Initially, early implementations of Secure Boot still allowed users to retain their traditional startup hotkeys. However, as operating systems like Windows evolved to maximize efficiency, a feature known as Fast Startup was introduced. Fast Startup altered the traditional shutdown process, effectively putting the kernel into a deep hibernation state rather than a cold power-off. Consequently, the POST phase was abbreviated to fractions of a second, or bypassed entirely, closing the traditional physical window required to register a keyboard input during startup.

As a result, modern systems shifted the primary mechanism for accessing the UEFI firmware away from the hardware initialization phase and directly into the Windows operating system itself. For the modern user, getting into the UEFI configuration requires navigating a deliberate sequence within the OS software interface. One must open the Windows Settings app, navigate to System, select Recovery, and choose **Advanced Startup** to initiate a specialized reboot. Only after the system cycles through a secondary blue environment—requiring the user to click through "Troubleshoot," then "Advanced Options," and finally "**UEFI Firmware Settings**"—will the machine finally restart into its underlying motherboard configuration interface. This design effectively treats the fundamental firmware of your own motherboard as an applet managed and permitted by Windows, a significant departure from independent hardware control.

The complexity deepens further on many modern pre-built desktops, laptops, and enterprise-focused motherboards. To protect users from unauthorized hardware alterations, manufacturers have implemented strict validation rules within the UEFI layout. On several modern systems, even if you successfully navigate through the Windows Advanced Startup menus to enter the UEFI, you are greeted with a locked or severely restricted configuration environment. To unlock the full suite of settings, override boot restrictions, or permanently restore the ability to use the traditional Delete or F2 keys directly on a cold reboot without going through Windows, the system requires the creation or validation of a **Supervisor Password** within the firmware.

This introduces a frustrating paradox: to reclaim a classic hardware-level convenience, the user must implement an additional layer of local encryption security. Setting a supervisor password tells the UEFI firmware that an authorized administrator is explicitly managing the device's physical profile. On some platforms, this action satisfies the hardware's internal security matrix, allowing the firmware to reopen the brief boot-time polling

window for external keyboard commands. The user is forced to trade a password-free local environment just to regain the baseline tactile accessibility that used to be a universal default.

However, the consequences of setting this firmware password or altering the Secure Boot state to reclaim direct keyboard access ripple directly back into the Windows ecosystem, creating a cascading security failure for user authentication. Modern iterations of Windows utilize the **Trusted Platform Module (TPM)**, a specialized physical chip, **or firmware-based architecture**, to handle cryptographic keys for biometric data and local sign-in shortcuts. Features like the **Windows Hello PIN** do not store your account password locally; instead, they rely on a cryptographic container bound directly to the unique, uncompromised hardware state of the machine.

When a user enters the UEFI, sets a supervisor password, alters configuration properties, or forces an alternative boot path to bypass Windows intervention, the structural signature of the platform shifts. The next time the computer initializes, the TPM evaluates the platform configuration registers. Recognizing that a **foundational change** has occurred at the firmware level, the TPM interprets the modification as a potential physical tampering attempt or an unauthorized hardware exploitation vector. To prevent data extraction, the TPM immediately invalidates the stored cryptographic keys associated with the Windows Hello container.

The practical reality of this interaction is a jarring lock-screen experience for the user. Upon booting back into Windows after managing their UEFI settings, the user finds their convenient, short PIN completely disabled. The system displays an error indicating that something changed with the security profile, forcing an immediate rollback to the primary account security layer. Windows completely bypasses biometric and PIN options, demanding that the user manually enter their full, complex master account password instead.

For many users who have relied entirely on a four-digit PIN or a fingerprint for months or years, this sudden reversion introduces an immense hurdle, frequently trapping them in account recovery loops if the long-form master password has been forgotten. This circular problem perfectly captures the frustrating state of modern system administration: an attempt to modify firmware settings forces the user to navigate an OS-controlled menu, which requires setting a hardware password, which ultimately breaks the OS login token, forcing a return to legacy passwords.

Ultimately, what began as an industry-wide effort to defend users against malicious boot exploits has evolved into a system that routinely treats the legitimate hardware owner with suspicion. The simple act of mashing a keyboard key to tell a motherboard which drive to read was a hallmark of personal computing freedom. The contemporary reality—where firmware access must be negotiated through Windows, gatekept by newly enforced supervisor passwords, and penalized by broken login PINs—demonstrates how much control has been surrendered. Security is undeniably vital in a hyper-connected world, but when the architecture of a computer becomes so heavily engineered that an owner must solve a multi-layered cryptographic puzzle just to view their own system settings, the line between protecting the user and locking them out of their own device becomes remarkably thin.

North Orange County Computer Club

Dr. Donald Armstrong
709 Rosarita Drive
Fullerton, CA 92653

To All Members:

The line above your mailing address
now shows your joindate. Please use
your join **month** to choose when to
renew your membership.

Dated Material – Please deliver ASAP

Membership Level (\$)	1 Year	3 Years
Individual Member	35	90
Each Additional Family Member	15	40
Full-Time* Enrolled College Student	20	
Enrolled High School Student	15	
*Minimum 12 Semester Hours		

Business Member + Ad (Business Card)	25	
Business Member + Ad (¼ Page, ½ Page)	65,	100
Business Member + Ad (Full Page)	175	
Contributing Member	75	
Supporting Member	100	
Advocate Member	250	
Patron Member	500	

Directions to the NOCCC meeting location



Enter CA-55 N (Costa Mesa Freeway) crossing Interstate 5 toward Anaheim/Riverside for 9 miles. *Notice freeway and street signs stating "Chapman University."* Exit toward E Chapman Ave. Turn right onto N Tustin St. Turn left onto E Walnut Ave.

1) Turn left past N. Center St. for the **best place to park** in the underground parking structure (Lastinger under the sports field). Pay the small fee (\$2) to park Ask members or help@noccc.org about parking details, restrictions, and our price break!

2) Turn left onto N Center St. On the right is the Hashinger Science Center, 346 N Center St. Orange California. Parking on the University side is free. Parking on the residential side is a city violation that may cost you a **tow away and a ticket!**