Chapter 11 – pasted from instructions → OS 464------All reference items are from your Textbook (5th Edition).
With more than 200 words and less than 15% similarity rate is required.

- Regarding the Sysinternals suite available at Microsoft Sysinternals, please identify two specific tools ("Process Explorer" link:https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer is the required one.) that pique your interest for in-depth exploration. As a student of computer science, could you articulate your perspective on the potential value these tools may provide in your academic and future professional development?
- (p. 1036) question#1
- (p. 1039) **Explain!!!** question#37
- (p. 1040) question#46

---

1) 1st bullet point ans

One of the most used tools for the microsoft sysinternals suite is process explorer. This tool is capable of going beyond the normal capabilities of task manager by offering deeper visibility into current running processes, the handles and DLLs(dynamic link library) they've loaded and their memory usage as well. I see this tool as an opportunity to understand how windows manages its applications, dependencies and system resources. For instance while working on a debugging project. The process explorer can help identify memory leaks, trace back problematic DLL versions, or find and detect unauthorized processes that may be running in the background. These features are very helpful especially when working directly with OS concepts/projects and security.

Another tool I find very interesting is Autoruns, which has the ability to display all the currently configured programs to run at computer startup. This includes the startup folder items and run registry keys but also browser helper objects/tools, scheduled tasks and drivers. A career field where this is very helpful is cybersecurity where understanding the system persistence is very important. Autoruns provides detailed information into how software integrates with the OS at boot. It can also be utilized in performance tuning by selecting which startup items to run or not to run. Together these tools give the user a deeper look into the system internals and how they function with one another.

2) **1. Give one advantage and one disadvantage of the registry vs. having individual *.ini* files.**

An advantage of the registry over individual ".ini" files is that the registry can provide a centralized and more secure database/repository for system and application settings. Unlike .ini files, which are located across different directories and file systems, the registry stores configurations in a structured style with access controls, reducing the risk of unauthorized modifications and improving management. This centralized design also enables faster access and querying of configuration data, which is beneficial for performance-sensitive applications. However, the registry also has a significant drawback: As it lacks the portability and simplicity of .ini files. This is one of its major downfalls. Since registry data is stored in a binary format and wired to a specific system which it cannot be removed from, it cannot be easily edited, transferred, or backed up like plain-text .ini files could be .

Additionally, a disadvantage of the registry is corruption. This can lead to serious system-wide issues, whereas a damaged .ini file typically only affects only a single application(usually). Furthermore registry edits also require more technical domain knowledge and caution, as incorrect changes can destabilize the entire operating system. While the registry enhances security and organization, .ini files remain more user-friendly and robust for manual configuration, especially in lightweight or portable software scenarios.

3) **37) Suppose that you wanted to build Windows Lite. Which of the fields of Fig. 11-55 could be removed without weakening the security of the system?**

After reviewing the figure in the book of the access token structure. I found some components are required while others are not, for instance User SID is required, it is the item that can uniquely identify the user who runs the process and is central to any access token decision. The groups field is also required as it determines what collective permissions the user is able to inherit. The privilege field must also stay as it is responsible for what system level actions the token holder cna perform. And the header provides structure and versioning so those items cannot be removed.

On the other hand, fields such as expiration time are more related to session management and are not commonly enforced in standard windows tokens. In theory removing it would have little impact (unless you're implementing token timeouts -which a lite OS most likely will not require). The default DACL provides default permissions for objects the process creates in a simplified OS, explicit ACLs can be applied manually when required, making this item optional.

Restricted SIDs are useful typically in high-security environments but not required in a lightweight system focused on basic functionality. The impersonation level, mostly used in client-server communications to determine how much authority a process can inherit, can be excluded if impersonation features are not supported. Finally, Integrity Level, used in mandatory integrity control(MIC) to isolate processes based on trust, is also optional if MIC is not implemented.

In summary, expiration time, default DACL, group SID (if redundant with groups), restricted SIDs, impersonation level, and integrity level can be removed in a windows lite version to run the system without compromising its foundational security principles.

4) **6. Windows 10 introduced a feature known as Microsoft Defender Application Guard that allows the Edge browser and Microsoft Office apps to run a hardware isolated con-tainer, and remotes the UI back to the host. The result is that the application appears to the user to be running locally even though it's actually hosted in a type of VM. What subtle user experience issues could this cause?**

Microsoft Defender Application Guard increases security by running applications like microsoft edge and office in a hardware-isolated container, protecting the host system from potentially malicious content encountered during browsing or file access. This isolation is a powerful defense feature, especially in enterprise or large scale environments where users often interact with unknown websites or external documents. However, while this approach greatly enhances protection, it introduces some subtle user experience issues. One common problem is inconsistent application behavior compared to the standard environment. Because these applications are running in a virtualized container, they may behave differently from their non-isolated counterparts in some cases. For instance, users might notice that certain settings or preferences they have configured previously outside the container such as browser bookmarks, saved passwords, or interface themes do not carry over within the container.

This disconnect can lead to confusion, as users typically expect a consistent experience across all versions of an application. In addition, integration with other system features, such as single sign-on or third-party add-ins, may be restricted or unavailable within the container. File access and clipboard functionality can also be limited, further disrupting workflows/user experience. While the security benefits of microsoft defender application guard are impressive, these usability trade-offs must be carefully weighted to maintain productivity and minimize user confusion, especially in environments where consistent interaction is expected.