

www.toutwindows.com



Vous trouverez dans ce document un récapitulatif des articles de toutwindows.com.

Les commentaires et corrections sont les bienvenus !

Version 1.0



Laurent Gébeau

www.Toutwindows.com

Pour me suivre :



Table des matières

Le boot de Windows 10 / UEFI	3
Etape 1 : Préboot UEFI.....	3
Etape 2 : Windows Boot Manager	3
Etape 3 : Windows Loader	4
Etape 4 : Noyau Windows	4
Etape 5 : Ouverture de session	5
smss.exe : session manager subsystem.....	6
win32k.sys : multi user win32 driver file	8
csrss.exe : Client Server Runtime Subsystem.....	9
winlogon.exe : Windows Logon Application	10
wininit.exe : Windows Start-Up Application.....	11
lsass.exe : Local Security Authority Subsystem	12
LogonUI.exe : Windows Logon User Interface Host.....	13
services.exe (Service Control Manager)	15
svchost.exe : Host Process for Windows Services	16
userinit.exe : Userinit Logon Application	18
explorer.exe (Explorateur Windows).....	19

Le boot de Windows 10 / UEFI

La séquence de démarrage de Windows 10 (boot) est très précise et sécurisée, je vous propose de la découvrir en détails.

Lorsque votre PC ne démarre plus, cela vous aidera à vous y retrouver et à utiliser le bon outil.

Etape 1 : Préboot UEFI

Lorsque vous démarrez votre PC le premier code exécuté est le **UEFI** (de l'anglais **Unified Extensible Firmware Interface**, signifiant en Français : « Interface micrologicielle extensible unifiée »), digne remplaçant du BIOS, de l'anglais Basic Input Output System (en français : « système élémentaire d'entrée/sortie »).

L'UEFI démarre tout d'abord les tests du matériel (équivalent au POST en BIOS) puis recherche les informations de boot dans la SRAM.

Dans UEFI Secure Boot est un protocole qui permet de sécuriser le processus de chargement du système d'exploitation en reconnaissant (ou rejetant) une signature numérique associée à l'OS et au Firmware.

Pour en savoir plus, je vous conseille cet excellent article :

[UEFI Secure Boot: Who controls what can run? – Out of Office Hours \(oofhours.com\)](http://www.outofofficehours.com/uefi-secure-boot-who-controls-what-can-run/)

et celui-ci

https://docs.microsoft.com/en-us/windows-hardware/drivers/bringup/secure-boot-and-device-encryption-overview?WT.mc_id=WDIT-MVP-9999

Etape 2 : Windows Boot Manager

Le gestionnaire de boot Windows (Windows Boot Manager), qui se trouve dans **EFI\BOOT\BOOTX64.EFI** est lancé.

Le rôle du Windows Boot Manager est de lancer le système d'exploitation, il le trouve dans le fichier BCD (Boot Configuration Data) qui se trouve dans /EFI/Microsoft/Boot/BCD.

On peut consulter les entrées de démarrage dans l'UEFI ou à l'aide de la commande BCDedit :



BCDedit :

```
ca. Administrateur : Invite de commandes

C:\WINDOWS\system32>bcdedit

Gestionnaire de démarrage Windows
-----
identificateur      {bootmgr}
device              partition=\Device\HarddiskVolume2
path                \EFI\Microsoft\Boot\BootMGFW.EFI
description          Windows Boot Manager
locale              fr-FR
inherit              {globalsettings}
default              {current}
resumeobject         {e66dfc11-f4e1-11e8-87f2-c21a747019bc}
displayorder         {current}
toolsdisplayorder    {mendiag}
timeout              30

Chargeur de démarrage Windows
-----
identificateur      {current}
device              partition=C:
path                \WINDOWS\system32\winload.efi
description          Windows 10
locale              fr-FR
inherit              {bootloadersettings}
recoverysequence     {e66dfc14-f4e1-11e8-87f2-c21a747019bc}
displaymessageoverride Recovery
recoveryenabled       Yes
testsigning          No
isolatedcontext       Yes
allowedinmemorysettings 0x15000075
osdevice             partition=C:
systemroot           \WINDOWS
resumeobject         {e66dfc11-f4e1-11e8-87f2-c21a747019bc}
nx                   OptIn
bootmenupolicy        Standard
hypervisorlaunchtype Auto

C:\WINDOWS\system32>
```

Etape 3 : Windows Loader

Le BCD pointe vers le Windows loader (Winload.exe). Celui-ci se trouve par défaut dans **%SystemRoot%\system32\winload.efi**

Le Windows Boot Loader peut désormais lancer le noyau de Windows.

Etape 4 : Noyau Windows

Le noyau de Windows se trouve ici :

%SystemRoot%\system32\ntoskrnl.exe

ntoskrnl.exe (NT operating system kernel executable), souvent appelé kernel image, contient le noyau de Windows

L'exécution de celui-ci commence :

- inventaire matériel
- chargement des drivers de base (ceux qui sont identifiés comme BOOT_START dans le registre)
- le contrôle est ensuite passé à smss.exe

smss.exe est nommé gestionnaire de session il lance :

- la gestion d'un certain nombres d'opérations d'initialisation : système de fichiers, périphériques DOS, ouvre la session 0 (services) et 1 (utilisateur), la mémoire virtuelle
- Les sous systèmes du noyau (**win32k.sys**)
 - Plus d'informations ici : [smss.exe session manager subsystem](#)

win32k.sys en charge du gestionnaire de fenêtres et des services GDI lance les processus suivants :

- les services démarrent
- le mode utilisateur **csrss.exe** (*Client/Server Runtime SubSystem*) : qui gère la console et l'interface
- le process de gestion du profil utilisateur est lancé (**winlogon.exe**)
- **Winlogon.exe** démarre et affiche l'écran d'ouverture de session,
- winlogon lance **%WINDIR%\System32\lsass.exe** (*Local Security Authority Subsystem*) qui est le système chargé de l'authentification (session, mots de passe, tokens, Kerberos, IPSec, AD) et qui écrit les événements de sécurité.

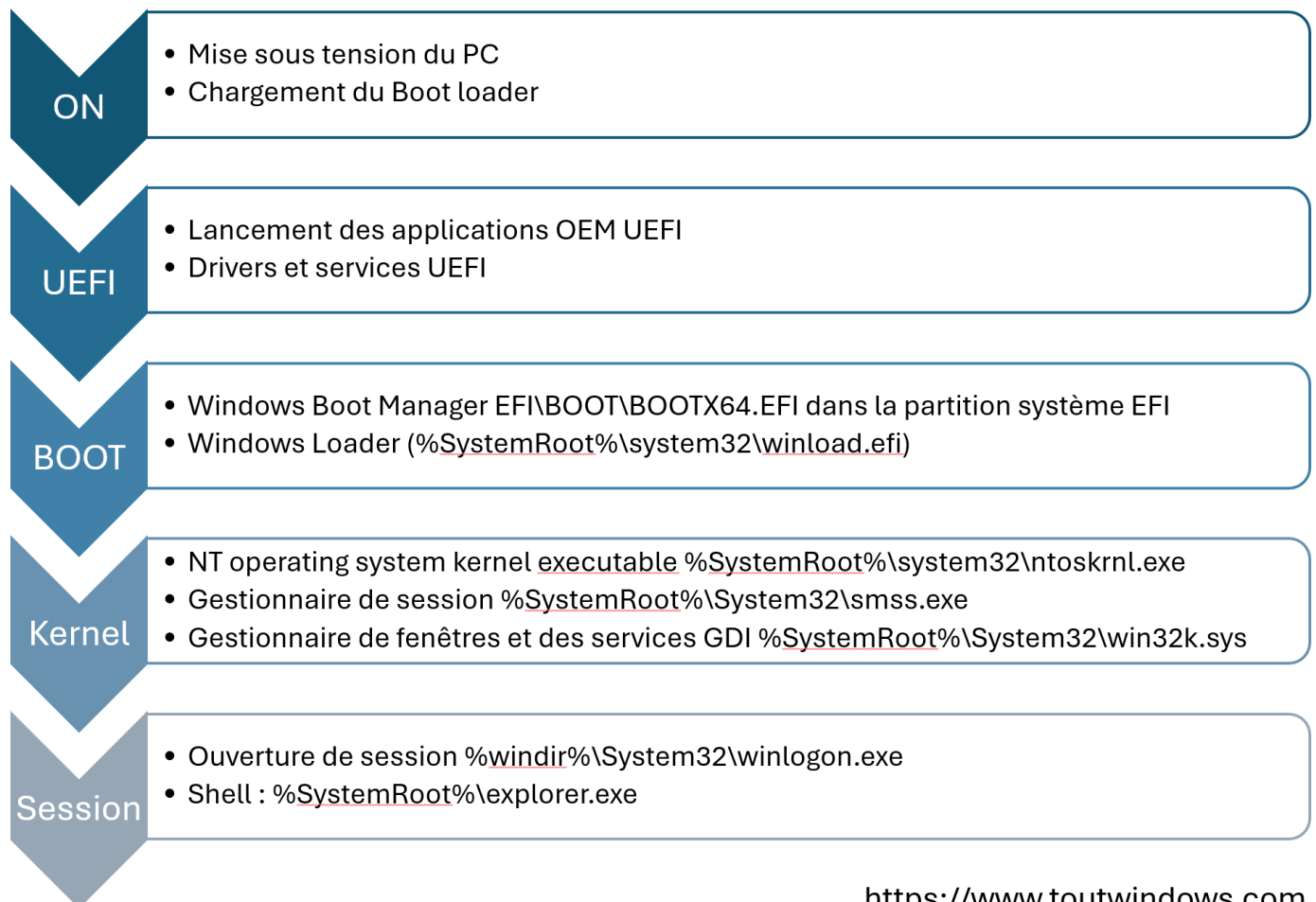
Etape 5 : Ouverture de session

Les services continuent de se charger

les GPO sont exécutés

La session est ouverte et le shell est lancé (explorer.exe)

Résumé :



smss.exe : session manager subsystem

Je vous ai déjà expliqué comment se passe le démarrage de Windows en détail.

Je vous propose désormais de vous détailler le rôle des process principaux de Windows, et tout d'abord le premier qui se nomme **smss.exe** mais aussi **Session Manager Subsystem**.

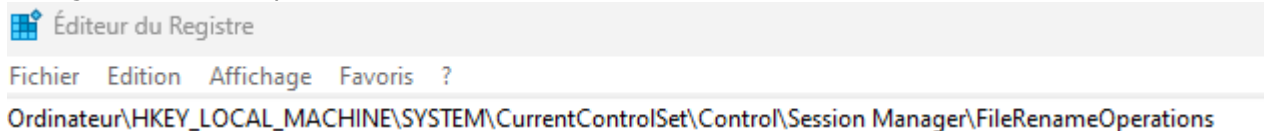
Ce fichier se trouve dans %SystemRoot%\System32\smss.exe

Il s'agit du premier process lancé dans le contexte utilisateur.

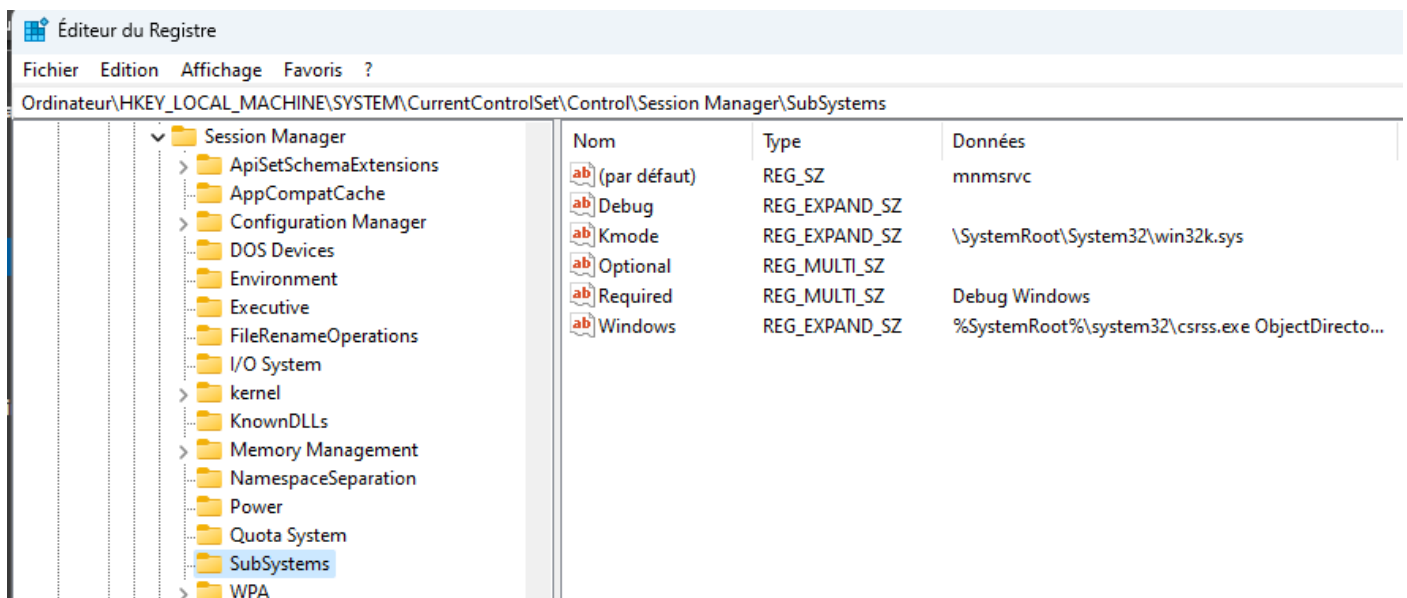
Celui ci va gérer plusieurs opérations de démarrage

- test du système de fichiers (autochk.exe)
- renommage (ou suppression) des fichiers nommés dans le registre à la clef "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session

Manager\FileRenameOperations"

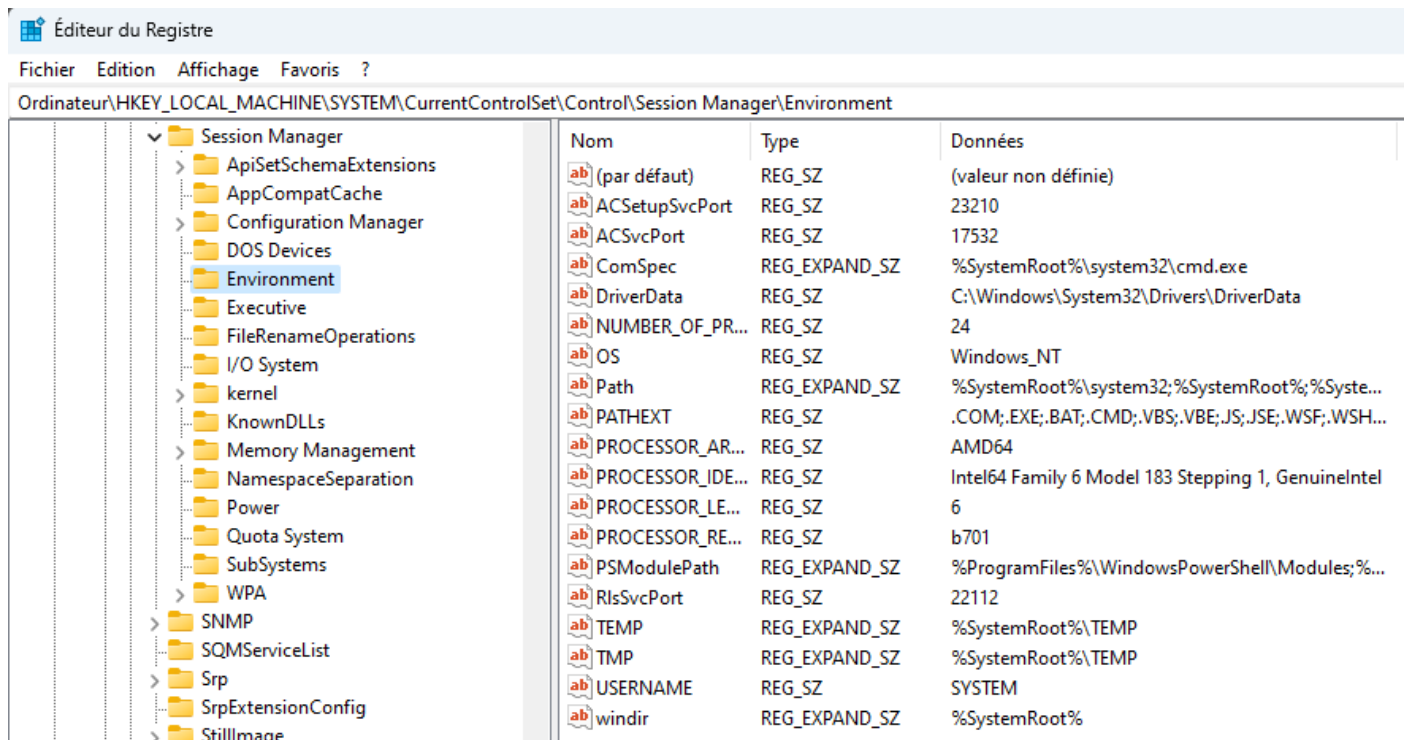


- création des périphériques DOS (AUX, CON, PIPE...). Ces périphériques sont créés sous forme de liens symboliques (symbolics links) dans les objets systèmes 32 bits.
- chargement des sous systèmes déclarés dans "HKLM\System\CurrentControlSet\Control\Session Manager\SubSystems".



- le sous système 32 bits du noyau Windows : \SystemRoot\System32\[win32k.sys](#)
- La session 0, qui est la session dans laquelle les services sont exécutés est ouverte en lançant %SystemRoot%\system32\csrss.exe csrss.exe (Client Server Runtime Subsystem) et %windir%\System32\wininit.exe (Windows Start Application)
- La session 1, qui est la première session utilisateur est ouverte en lançant %SystemRoot%\system32\csrss.exe csrss.exe (Client Server Runtime Subsystem) et %windir%\System32\winlogon.exe (Windows Logon Application)

- Le fichier d'échange utilisé par la mémoire virtuelle est créé (avec les paramètres situés dans la clef HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory management)
- les variables d'environnement situées dans HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\Environment sont créées (%PATH%, %PATHEXT%, %TEMP%, %TMP%, %WINDIR%, %OS%, %COMSPEC% (voir (en) ComSpec, %NUMBER_OF_PROCESSORS, %PROCESSOR_ARCHITECTURE%, %PROCESSOR_IDENTIFIER%, ...etc.)



Pendant le fonctionnement de Windows smss.exe gère un certain nombre d'opérations :

la mémoire et le fichier d'échange, les DLL, la gestion de l'alimentation.

smss.exe est aussi utilisé lors de l'ouverture de session RDP.

smss.exe est affiché dans le gestionnaire des tâches sous le PID 4, nommé system, car il en est un sous process :

Voici la vue des process de Process Explorer des Sysinternals :

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
System	< 0.01	60 K	2256 K	4		
Interrupts	0.19	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1124 K	1256 K	884		
Memory Compression		1616 K	776880 K	5316		

Sources :

https://fr.wikipedia.org/wiki/Session_Manager_Subsystem

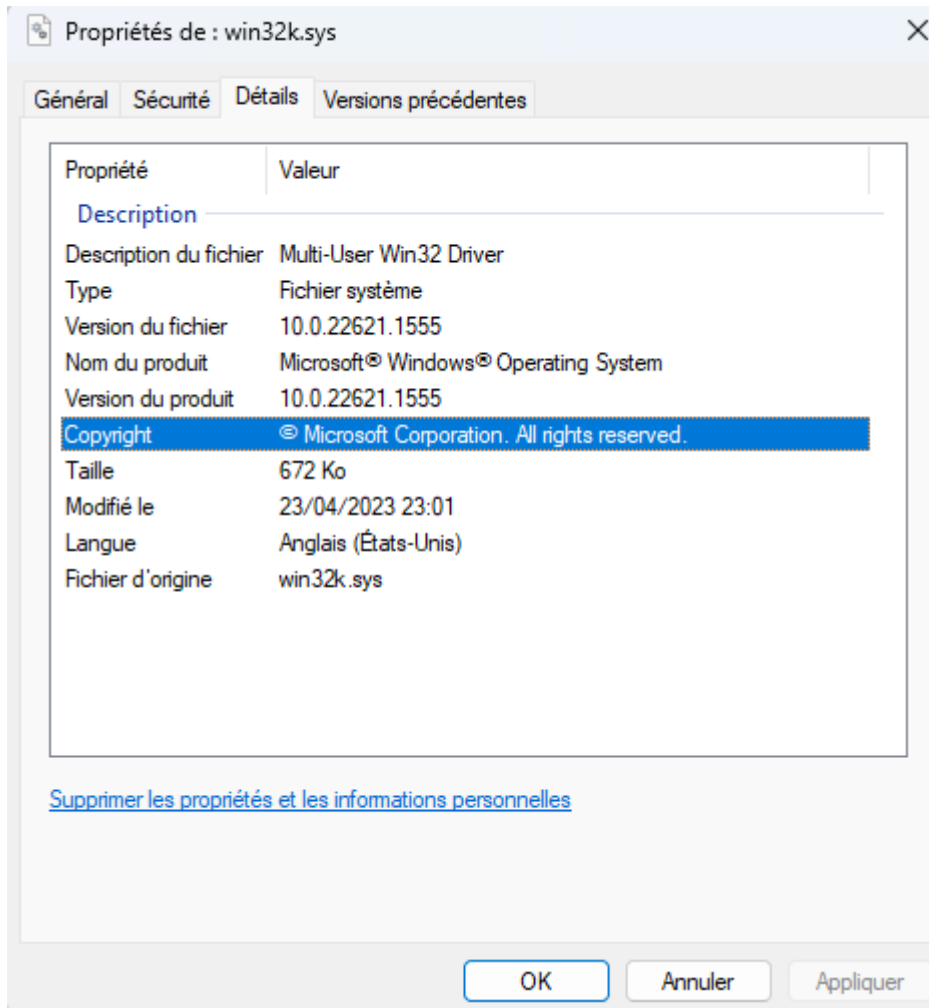
The Windows Process Journey By Dr. Shlomi Boutnaru – <https://medium.com/@boutnaru/the-windows-process-journey-smss-exe-session-manager-subsystem-bca2cf748d33>

<http://winapi.freotechsecrets.com/win32/WIN32DefineDosDevice.htm>

win32k.sys : multi user win32 driver file

Voici le moment de parler de **win32k.sys** aussi nommé **Multi User Win32 driver file**.

Ce fichier se trouve dans %SystemRoot%\System32\win32k.sys



C'est une interface qui parvient à envoyer des graphiques aux moniteurs et autres périphériques de sortie. Le code est exécuté par gdi32.dll sous Windows 10 et 11.

Il comprend les DLL de l'API Win32 et le processus du sous-système Win32 (csrss.exe).

- kernel32.dll : Client des API de base Windows, c'est lui qui les expose aux applications, leur permettant de gérer la mémoire, les E/S, les process et thread, la plupart de ces fonctions sont exécutées en appelant NTDLL.DLL
- user32.dll : il contient les fonctions de l'API Windows liées à l'interface utilisateur Windows (GUI), la plupart de ces fonctions sont exécutées en appelant GDI32.DLL
- gdi32.dll : interface graphique de périphérique (GDI) qui permet de fournir les fonctions de base graphiques de dessin de bas niveau (ligne, rectangle, ...), les polices et couleurs.
- csrss.exe : processus d'exécution du client/serveur responsable, entre autres, de la capacité de démarrage et d'arrêt de nombreux autres processus du système. Il gère également la ligne de commande.

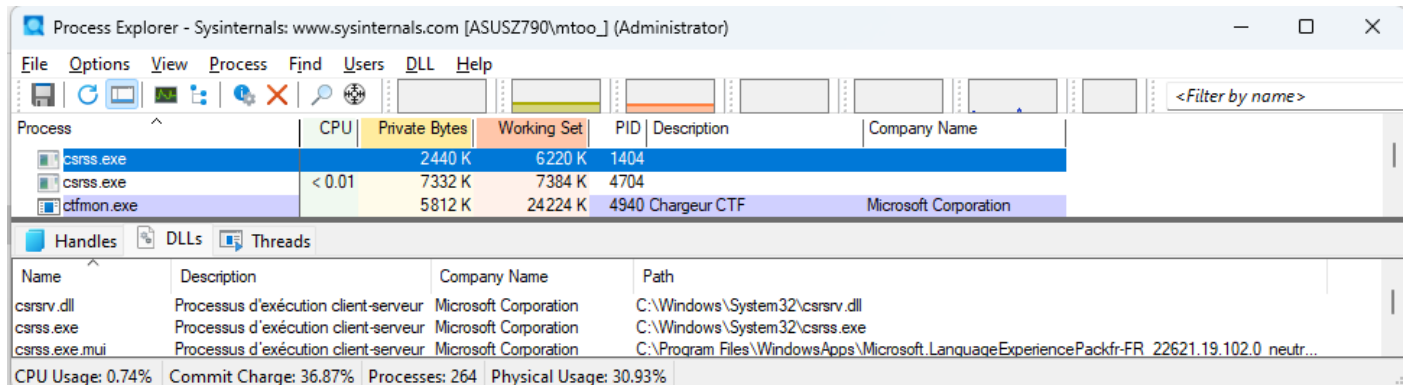
csrss.exe : Client Server Runtime Subsystem

Le process csrss.exe (Client Server Runtime Subsystem) est le process chargé d'initialiser la session utilisateur. Ce fichier se trouve dans %windir%\System32\csrss.exe".

csrss gère les process et les threads, il gère aussi les sessions CMD (conhost.exe) et les fichiers temporaires.

csrss est lancé dans le contexte « local system » et il existe une instance par session (la session systeme 0 étant la première, il y a donc minimum deux instances).

"CRSS.EXE" charge "csrsrv.dll", "basesrv.dll" et "winsrv.dll" .



The screenshot shows the Process Explorer window from Sysinternals. The 'Process' tab is active, displaying a list of running processes. Three instances of csrss.exe are visible, along with ctfmon.exe. The 'Handles' tab is also shown, listing the DLLs loaded by the selected csrss.exe process.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
csrss.exe		2440 K	6220 K	1404		
csrss.exe	< 0.01	7332 K	7384 K	4704		
ctfmon.exe		5812 K	24224 K	4940	Chargeur CTF	Microsoft Corporation

Name	Description	Company Name	Path
csrsrv.dll	Processus d'exécution client-serveur	Microsoft Corporation	C:\Windows\System32\csrsrv.dll
csrss.exe	Processus d'exécution client-serveur	Microsoft Corporation	C:\Windows\System32\csrss.exe
csrss.exe.mui	Processus d'exécution client-serveur	Microsoft Corporation	C:\Program Files\WindowsApps\Microsoft.LanguageExperiencePackfr-FR_22621.19.102.0_neutr...

CPU Usage: 0.74% | Commit Charge: 36.87% | Processes: 264 | Physical Usage: 30.93%

Plus d'infos ici :

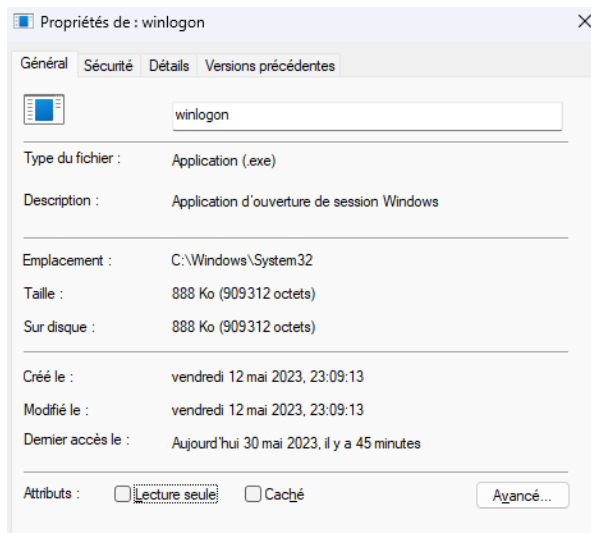
<https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/debugging-csrss>

winlogon.exe : Windows Logon Application

Continuons la découverte des process de Windows, découvrons maintenant winlogon.exe (Windows Logon Application).

winlogon.exe est le process chargé d'initialiser la session utilisateur.

Ce fichier se trouve dans %windir%\System32\winlogon.exe est lancé en tant que tâche système (owner: NT AUTHORITY\SYSTEM" (S-1-5-18)).



Ce programme gère les interactions d'authentification graphiques : le process d'ouverture et de fermeture de session, le lancement de l'interface graphique LogonUI.Exe, et communique avec lsass.exe.

La phase d'initialisation (Ctrl + Alt + Del = SAS : Secure Attention Sequence) est interceptée par winlogon afin d'éviter qu'un autre programme se l'accapare. WINLOGON.EXE crée trois bureaux :

- le bureau WinLogon desktop, pour présenter l'ouverture de session ou les demandes UAC
- le bureau DefaultDesktop ou Application Desktop, pour la session utilisateur
- le bureau ScreenSaverDesktop, réservé à l'écran de veille

WinLogon communique aussi avec la Graphical Identification and Authentication DLL. GINA est chargée par Winlogon qui implémente la stratégie d'authentification et d'ouverture de session et ses interactions graphiques.

Il y a 3 états de connexion (Winlogonstate) :

- Déconnecté (Logged-Off State)
- Connecté (Logged-OnState) qui est actif lorsque l'utilisateur a fourni les bons credentials et qu'il est autorisé à ouvrir la session
- Verrouillé (Workstation-Locked State) la session est ouverte mais est cachée par un bureau sécurisé, jusqu'à déverrouillage

Plus d'infos ici :

<https://learn.microsoft.com/en-us/windows/win32/winstation/desktops>

<https://learn.microsoft.com/en-us/windows/win32/secauthn/initializing-winlogon>

<https://learn.microsoft.com/en-us/windows/win32/secauthn/winlogon-states>

wininit.exe : Windows Start-Up Application






wininit.exe (Windows Start-Up Application).est le process chargé d'initialiser la session utilisateur. Ce fichier se trouve dans %windir%\System32\wininit.exe".

Wininit.exe est lancé par le premier process (session 0) de [smss.exe](#) sous le compte LocalSystem (S-1-5-18).

Une seule session de wininit.exe peut exister, wininit.exe de plusieurs étapes d'initialisations :

- création de %windir%\temp
- création de l'environnement du scheduler
- création des bureaux Winlogon et Default pour la session 0
- lancement de services.exe (Service Control Manager)
- lancement de lsass.exe (Local Security Authority Subsystem), depuis Windows10 lsalso.exe qui est une version conteneurisée de lsass.exe (processus en mode utilisateur isolé (IUM) dans un nouvel environnement de sécurité appelé vsm (Virtual Secure Mode)).
- lancement de fontdrvhost.exe" (Usermode Font Driver Host) pour la session 0

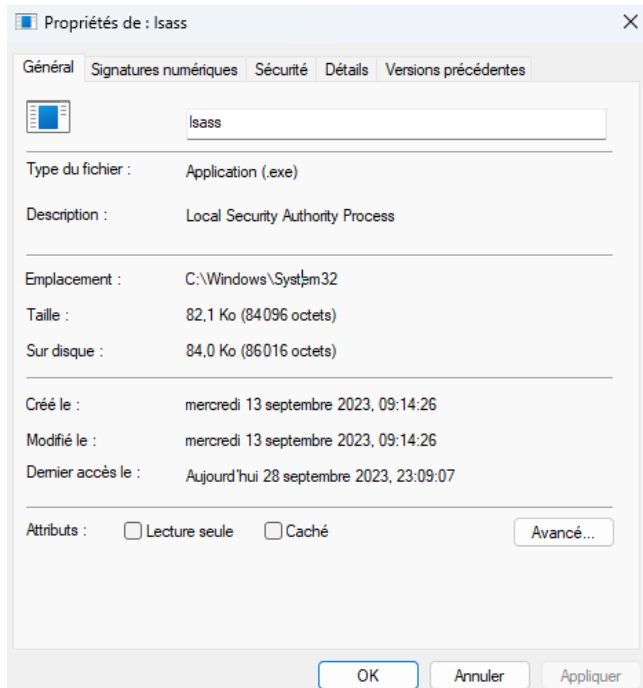
Ce process est marqué critique, Windows s'arrête si celui-ci est arrêté ou altéré.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
 wininit.exe		1624 K	6956 K	1436		
 services.exe	< 0.01	7696 K	17724 K	1668		
 lsalso.exe		2112 K	4828 K	1688		
 lsass.exe	< 0.01	15356 K	35116 K	1700	Local Security Authority Proc...	Microsoft Corporation
 fontdrvhost.exe		1960 K	5096 K	1852		

Isass.exe : Local Security Authority Subsystem

Isass.exe (Local Security Authority Subsystem).est un process qui gère la sécurité de Windows. Il fait partie du process Local Security Authority (LSA).

Il se trouve dans %WINDIR%\System32\lsass.exe.



Il vérifie la connexion a Windows (locale et distante), gère les changements de mot de passe, et attribue les tokens.

Isass.exe gère le journal de sécurité (visible dans l'observateur d'évènements).

En raison de son importance ce process est attaqué ou copié par les développeurs malveillants. Ce process est surveillé par Windows et si il n'est plus actif, Windows s'éteint.

Au démarrage de Windows, wininit.exe lance lsass.exe, de nombreuses vérifications sont effectuées autour de lsass.exe, par exemple les plug-in (lecteur de carte d'authentification, cryptographie, gestionnaires de mot de passe) l'entourant doivent être signés. UEFI peut apporter des protections supplémentaires pour l'isoler (UEFI lock) notamment à l'aide de HVCI.

Sources :

<https://learn.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/configuring-additional-lsa-protection>

LogonUI.exe : Windows Logon User Interface Host

LogonUI.exe (Windows Logon User Interface Host) est le process qui gère l'affichage de l'interface graphique de connexion utilisateur (logon screen / lock screen).

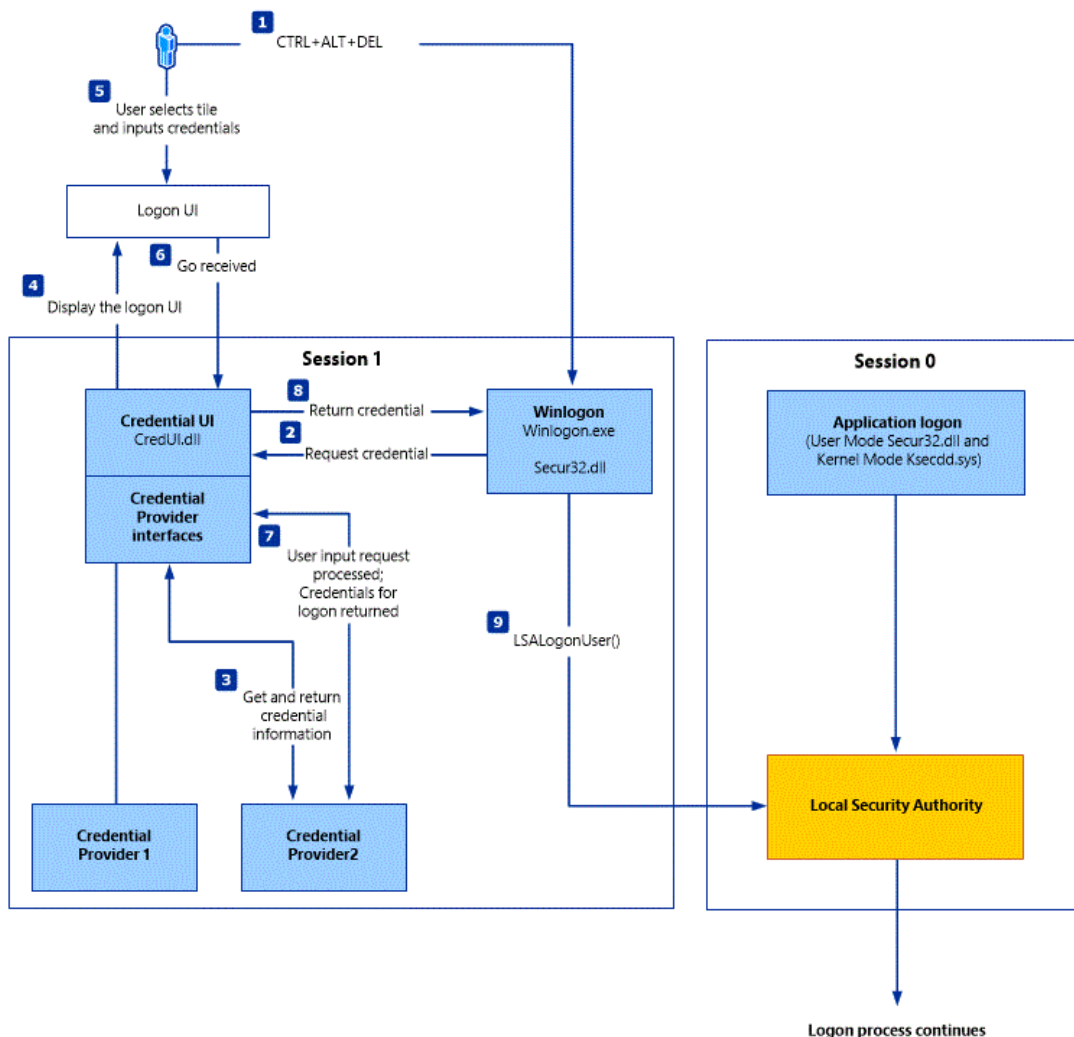
Il se trouve dans %WINDIR%\System32\logonUI.exe.

Ce programme est lancé par [winlogon.exe](#) par le compte Local System (S-1-5-18).

Voici le process résumé :

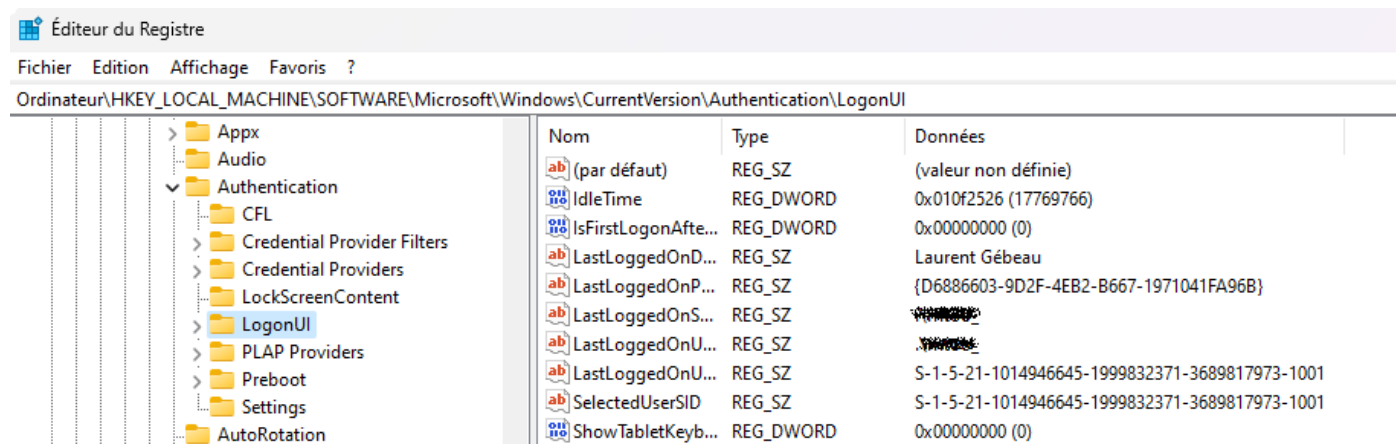
- Winlogon lance LogonUI.exe via le Credential UI (credui.dll)
- LogonUI.exe récupère les informations d'identification saisies par l'utilisateur (credentials).
- LogonUI.exe envoie des credentials à winlogon.exe à l'aide d'un fournisseur d'informations de connexion (credential provider)
- Winlogon.exe confirme l'authentification à LogonUI.Exe
- Winlogon transmet les credentials au Local Security Authority (LSA) via Secur32.dll.

En détails :



Dans le registre la branche

"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI" contient la liste des utilisateurs à afficher au logon, le dernier utilisateur connecté et l'image de fond.

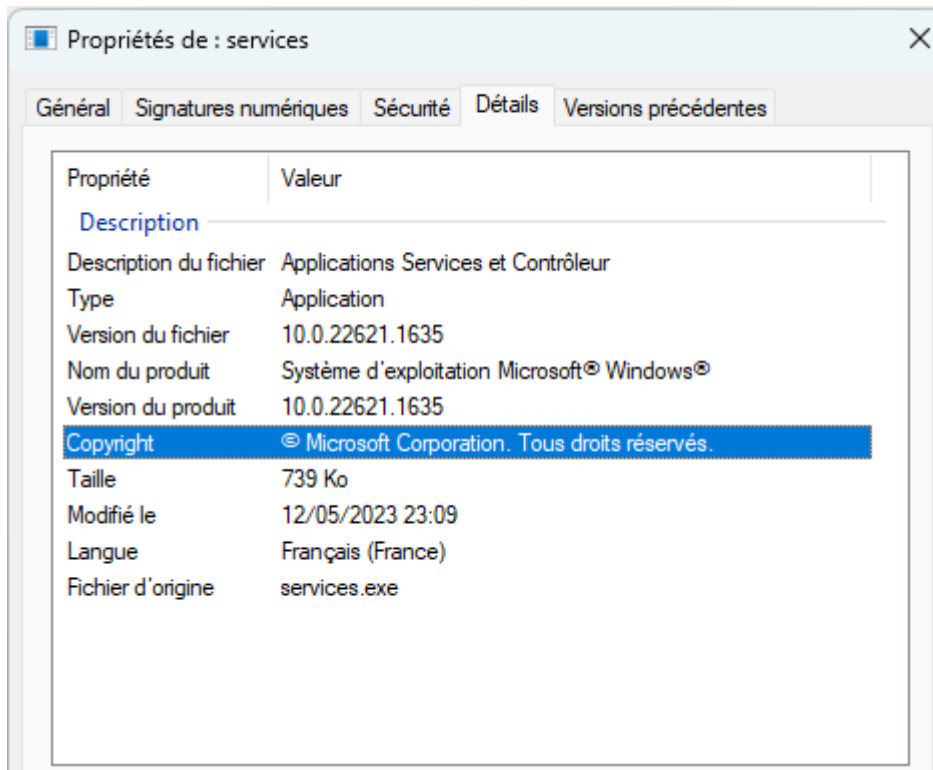


Le fonctionnement du process d'authentification en détail est expliqué ici :

<https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication>

services.exe (Service Control Manager)

services.exe (Service Control Manager) est le process qui gère les services Windows.



Celui-ci :

- gère la liste des services installés (autonomes, liés ou liés a un driver) dans le registre à l'emplacement suivant :
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

- démarre les services (soit au démarrage, soit en différé, soit lorsqu'un appel déclenche le service). L'ordre de démarrage est stocké dans le registre à l'emplacement suivant :
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ServiceGroupOrder\List

- surveille l'état de fonctionnement des services : démarré, en cours de démarrage, ou arrêté

- journalise les évènements de services

- prévient l'explorateur Windows lorsqu'un lecteur réseau est ajouté ou supprimé

Il se trouve dans %WINDIR%\System32\services.exe.

Ce programme est lancé par wininit.exe par le compte Local System (S-1-5-18).

Source : <https://learn.microsoft.com/fr-fr/windows/win32/services/service-control-manager>

svchost.exe : Host Process for Windows Services

svchost.exe (Host Process for Windows Services) est certainement le process le plus visible dans le Gestionnaire des tâches, car c'est lui qui est responsable d'héberger les services Windows.

La plupart des services étant des fichiers .dll, svchost est en charge de les exécuter en mémoire.

Processus	
Nom	Statut
Hôte de la fenêtre de la console	
> Hôte de service : groupe de services Unistack	
> Hôte de service : UtcSvc	
> Hôte de service : acquisition d'images Windows (WIA)	
> Hôte de service : Agent de stratégie IPsec	
▼ Hôte de service : appel de procédure distante (2)	
Appel de procédure distante (RPC)	
Mappeur de point de terminaison RPC	
▼ Hôte de service : Assistance IP	
Assistance IP	
▼ Hôte de service : Assistance NetBIOS sur TCP/IP	
Assistance NetBIOS sur TCP/IP	
> Hôte de service : BluetoothUserService_5e9c6	
> Hôte de service : CaptureService_5e9c6	
> Hôte de service : cbdhsvc_5e9c6	
> Hôte de service : CDPUserSvc_5e9c6	
> Hôte de service : Client de suivi de lien distribué	
> Hôte de service : Client DHCP	
> Hôte de service : Conteneur Microsoft Passport	
> Hôte de service : Découverte SSDP	
> Hôte de service : Détection matériel noyau	
> Hôte de service : DevicesFlowUserSvc_5e9c6	
> Hôte de service : Expérience audio-vidéo haute qualité Windows	
> Hôte de service : Générateur de points de terminaison du service Audio Wind...	
> Hôte de service : Gestionnaire de comptes web	

La version 64 bits se trouve dans %windir%\System32\svchost.exe
La version 32 bits se trouve dans %windir%\SysWOW64\svchost.exe.

Les paramètres des services se trouvent dans le registre dans
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services"

Éditeur du Registre			
Fichier Edition Affichage Favoris ?			
Ordinateur\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BITS\Parameters			
	Nom	Type	Données
	(par défaut)	REG_SZ	(valeur non définie)
	ServiceDll	REG_EXPAND_SZ	%SystemRoot%\System32\qmgr.dll
	ServiceDllUnloa...	REG_DWORD	0x00000001 (1)

Depuis Windows 10 chaque service possède sa propre instance de svchosts afin d'en accroître la sécurisation, avec deux exception :

- l'optimisation de la RAM consommée sur les PC ayant moins de 3,5 Go entraîne un regroupement des services.
- certains services ont besoin d'être groupés, ils peuvent être paramétrés ainsi à l'aide d'une clef de registre nommée SvchostSplitDisable

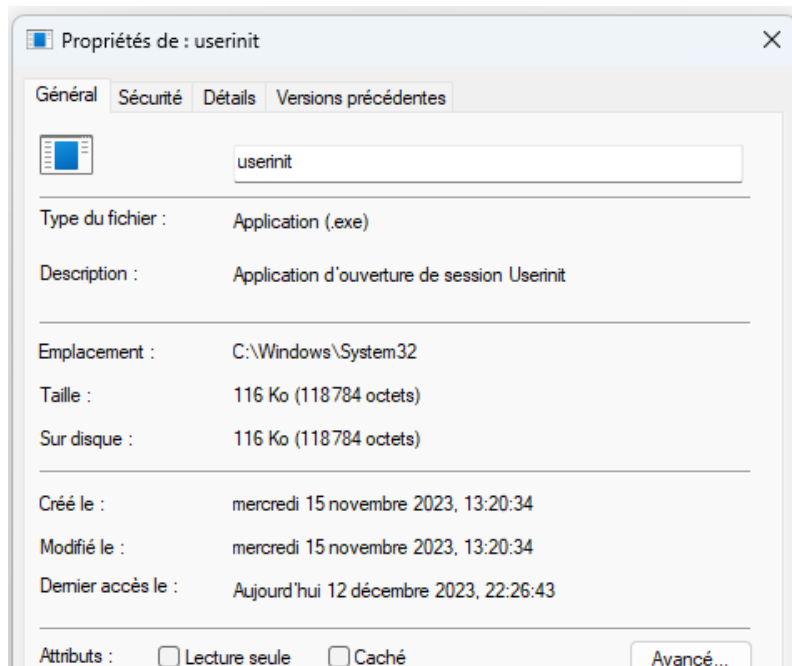
Source :

<https://learn.microsoft.com/en-us/windows/application-management/svchost-service-refactoring>

userinit.exe : Userinit Logon Application

userinit.exe (Userinit Logon Application) est le programme qui charge le profil utilisateur à l'ouverture de session. Celui-ci lance les programmes configurés au démarrage et exécute le login script. Une fois la session ouverte le process userinit est fermé.

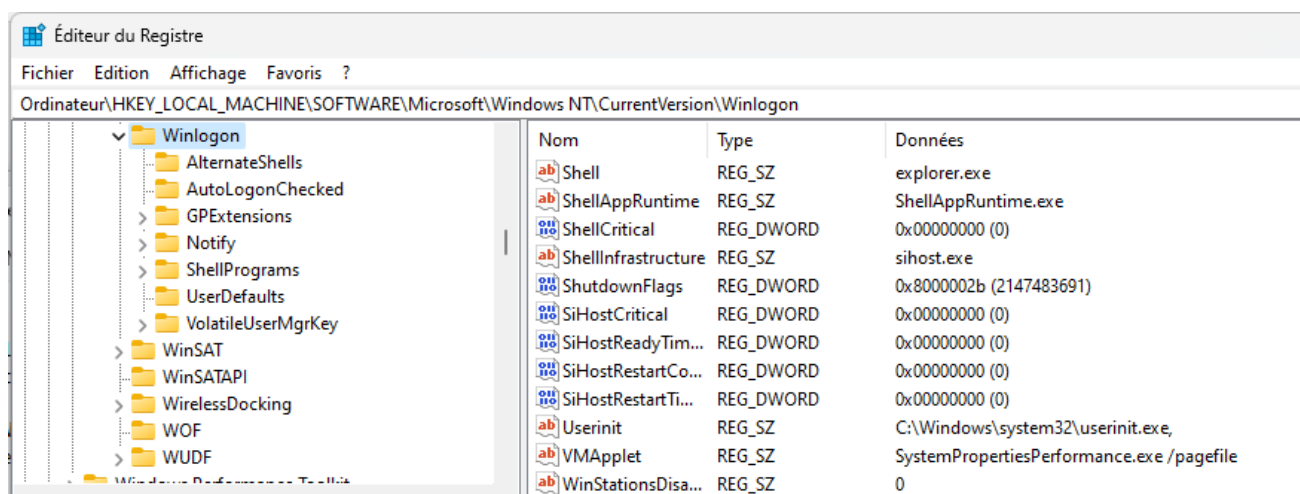
Il se trouve dans %WINDIR%\System32\userinit.exe.



Ce programme est lancé par [winlogon.exe](#) dans le contexte utilisateur.

Le programme déclaré comme responsable de l'initialisation (UserInit) par la clef de registre suivante :
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\userinit

userinit lance ensuite le programme déclaré comme Shell dans la clef suivante :
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\shell



explorer.exe (Explorateur Windows)

Explorer.exe est le programme gérant l'interface utilisateur. Techniquement explorer.exe est le shell de l'interface graphique. Dans la base de registre, ceci est défini par la clé HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\shell qui contient la valeur « explorer.exe ».

Explorer.exe charge l'explorateur de fichiers, mais aussi le menu démarrer et la barre des tâches. C'est aussi explorer.exe qui gère le bureau et la papier peint.

Pour fiabiliser Windows, Microsoft a séparé quelques process de explorer.exe :

- le menu démarré possède son propre process : StartMenuExperienceHost.exe

- la barre des tâches est désormais gérée par le process taskbar.dll,

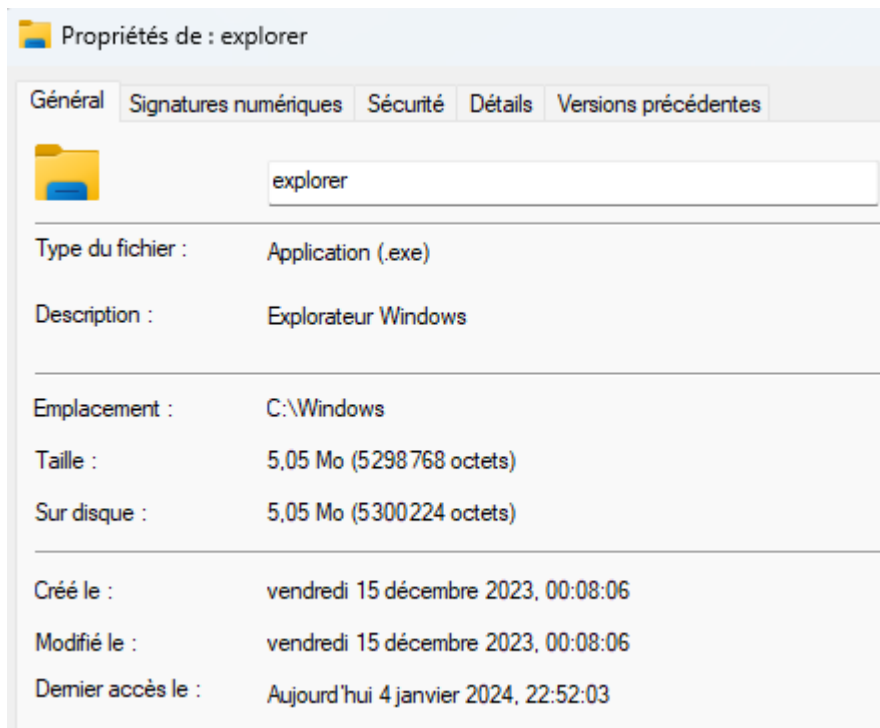
ceci permet par exemple d'éviter qu'un crash de la barre des tâche ne finisse par crasher l'ensemble des services rendus par l'explorer.

C'est userinit qui charge l'explorer.exe dans le contexte utilisateur.

userinit lance le programme déclaré comme Shell dans la clef suivante :

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\shell

Explorer.Exe se trouve dans %WINDIR%\explorer.exe.



Résumé

Voici un schema résumant le boot et les process de Windows :

