KATEDRA ZA SISTEMSKO INŽENJERSTVO I KIBERNETIČKU SIGURNOST

OPERACIJSKI SUSTAVI

Lab: Alati/Tools za pregled procesa Operativnog sustava

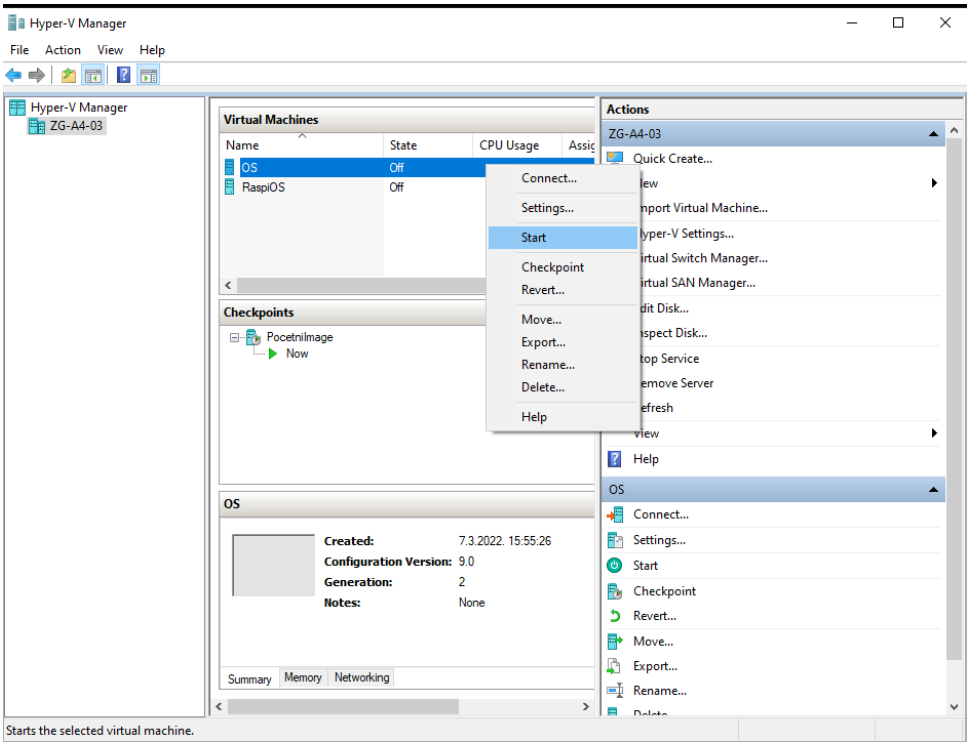
REV 2.2

Sadržaj

1 Pokretanje virtualnog računala

Za pokretanje virtualnog računala potrebno je:

1. Ulogirati se na studentsko računalo sa podacima:
    a. Korisničko ime: **hyperv**
    b. Korisnička zaporka/password: **hyperv**
2. Pokrenuti virtualno računalo putem Hyper-V managera i pokrenuti virtualno „**OS**“ računalo
3. Ulogirati se na virtualno računalo sa podacima:
    a. Korisničko ime/username: **korisnik**
    b. Korisnička zaporka/password: **Pa$$w0rd**



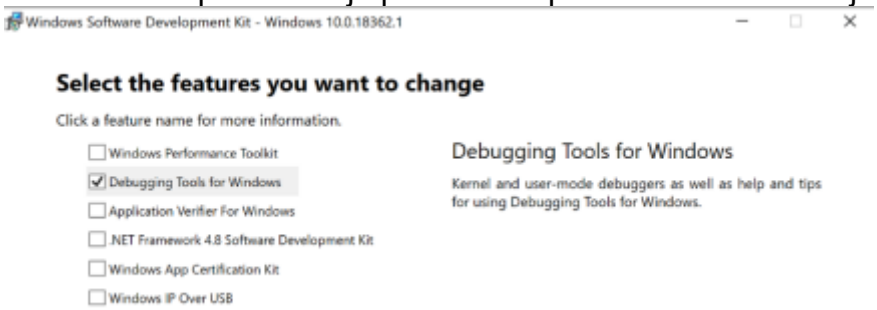
2 PID – Process ID

Jedno od svojstava procesa koje treba znati pronaći je **PID** procesa roditelj. Ova informacija je dobavljiva putem nekoliko alata, kao što su Performance Monitor, Task Manager, Proces Explorer (dio **Sysinternals** grupe alata) ili programski, putem funkcije **Create Process ID**.

U ovoj vježbi ćemo se upoznati s linijskim alatom **tlist.exe**, koji je dio **Windows Debugging Toolsa**, osnovnog alata za „duboko“ istraživanje (debug) Windowsa. Dotični alat se koristi, npr. kod razvijanja upravljačkih programa i možete ga besplatno preuzeti s Microsoftove web stranice.

Da bi koristili Debugging Tools for Windows (WinDbg) moramo preuzeti Windows 10 SDK s linka:
https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/

Prilikom instalacije ne morate instalirati sve komponente SDK već samo Debugging Tools for Windows. Nakon preuzimanja pokrenete i prođete kroz instalaciju.



2.1 Tlist.exe alat

1. Otvorite Command Prompt (**cmd**).
2. Pozicionirajte se u direktorij (naredba: cd putanja):
   C:\Program Files (x86)\Windows Kits\10\Debuggers\x64
   (Putanja može biti drugačija ako ste je mjenjali kod instalacije)
3. Upišite naredbu **tlist.exe /?**
4. Proučite ispis naredbe. Obavezno pročitajte opis prekidača **/t**.
5. Prikažite hijerarhijski procese korištenjem **/t** prekidača.
6. Ne zatvarajte Command Prompt

Napomena: Ukoliko je potrebno pokrenuti cmd u administratorskom modu – vidi Dodatak 1.

3 Veza među procesima

Uvlake prikazuju procese prema odnosu roditelj-dijete. Procesi čiji roditelji nisu „živi“ su uvučeni lijevo (pogledajte primjer s procesom **explorer.exe**) – zapravo, radi se o tome da Windowsi ne prate hijerarhiju procesa iznad razine roditelj-dijete. Windowsi bilježe jedino ID procesa roditelj a ne i ID procesa djed, pradjed itd. Demonstrirajmo:

3.1 Task Manager alat

1. Pokrenite **Task Manager** (Ctrl+Shift+Esc).
2. Prebacite se u **Command Prompt**
3. Upišite naredbu **mspaint**.
4. Prebacite se u Task Manager i kliknite na karticu **Details**.
5. Desnim gumbom miša kliknite na **cmd.exe** i odaberite opciju **End Process Tree**.
6. Kliknite na gumb **End Process Tree** na dijalogu upozorenja.
7. Primjetite da se i **mspaint** zatvorio.
8. Orvorite Command Prompt (**cmd**) i upišite naredbu **start cmd**. Ova naredba otvara novi Command Prompt u zasebnom prozoru.
9. Prebacite se u novootvoreni **Command Prompt**
10. Upišite naredbu **mspaint**.
11. Upišite naredbu **exit** (zatvaramo novi cmd).
12. Prebacite se u Task Manager
13. Desnim gumbom miša kliknite na **cmd.exe** i odaberite opciju **End Process Tree**.
14. Kliknite na gumb **End Process Tree** na dijalogu upozorenja.

-------- NAPOMENA --------

Prozor Command Prompta će nestati, ali MS Paint će ostati prikazan zato jer je to proces **djete** isključenog Command Prompta. S obzirom da smo proces roditelj MS Paint-a već isključili, relacija **djed-roditelj-dijete** je prekinuta i opcija End Process Tree nije mogla pratiti vezu od djeda do unuka.

15. Zatvorite sve otvorene prozore.

3.2 Process Explorer

Process Explorer je alat iz Sysinternals grupe. Dotični prikazuje vrlo detaljne informacije o procesima, detaljnije od svih ostalih alata.

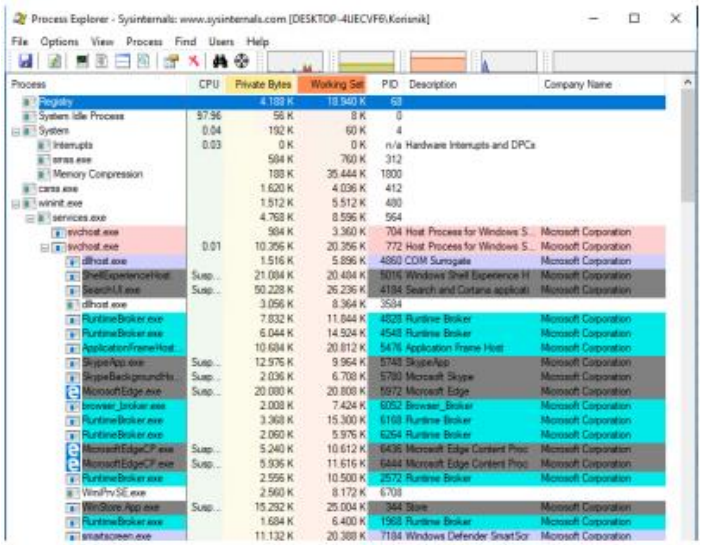
Navedimo njegove mogućnosti prikaza:

- Prikaz procesa koji su dio posla (eng. Job), kao i detalje o samom poslu
- Sigurnosni token procesa (popis grupa i njihovih privilegija na razini procesa)
- Bojanje procesa radi prikaza promjena u procesima i dretvama
- Vrijeme pokretanja procesa i dretvi
- Procese koje izvršavaju .NET aplikacije
- Kompletno mapiranje procesa na datoteke (ne samo dll datoteka, nego i ključeve Registryja, handleove itd.)
- Mogućnost suspenzije procesa
- Mogućnost prekidanja pojedinačnih dretvi unutar procesa
- Grafički prikaz resursa koje je proces zauzeo, kao i aktivnosti dretvi
- …

Da bi koristiti proces **explorer** moramo ga preuzeti s Microsoft stranica s linka: https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

Nije potrebno ništa instalirati već jednostavno raspakirate komprimiranu datoteku i pokrenete procexp.exe kao Administrator (**RUN-AS-ADMINISTRATOR**).

----------NAPOMENA----------
Ovaj alat, kao i velika većina drugih *Sysinternals* alata, zahtjeva prava lokalnog administratora.



Upoznajte se s alatom kroz sljedeće korake:

1. Ako je prikazano, isključite donje okno programa naredbom **View->Show Lower Pane**. Donje okno služi prikazu otvorenih handleova ili mapiranih DLL datoteka. S tim opcijama ćemo raditi na idućim vježbama.
2. Primijetite da su neki procesi obojeni **rozom** bojom. Ti procesi su zapravo servisi. Korisnički procesi su obojeni **plavom** bojom.
3. Postavite pokazivač miša iznad nekog procesa. Primijetite oblačić koji prikazuje putanju procesa do datoteke na disku, kao i dodatne informacije (npr. startup parametri).
4. Kliknite na **View->Select Columns**.
5. Odaberite **Image Path**.
6. **Sortirajte** prema stupcu **Process**. Primijetite da je nestao hijerarhijski prikaz. Procesi mogu biti sortirani prema hijerarhiji (predefinirano), te uzlazno ili silazno. Vratite hijerarhijski prikaz.
7. Kliknite na **View->Show Processes From All Users** kako bi prikazali samo svoje procese.
8. Kliknite na **Options->Difference Highlight Duration** i postavite vrijednost na 5 sekundi. Pokrenite novi proces (npr. MS Paint, kao u prvom dijelu vježbe). Primijetite da je taj proces obojen **zelenom** bojom. Isključite MS Paint i primijetite da je taj proces sad obojen **crvenom** bojom kroz 5 sekundi, prije nego se isključi. Ovakav vizualni prikaz je pogodan prilikom promatranja stvaranja i uništavanja procesa na računalu.
9. I za kraj, napravite dvostruki klik miša na neki korisnički proces i pregledajte sve kartice koje postoje na tom prozoru. Ove informacije će nam trebati na nekim idućim vježbama. Za sada, pokušajte **pronaći popis dretvi procesa**.
10. Zatvorite Process Explorer.

4 Kernel mod i korisnički mod

Na prijašnjim ste vježbama naučili da procesor i, samim time, operacijski sustav razlikuje dva načina rada – kernel i korisnički način. Vrijeme provedeno u tim načinima rada se može pratiti pomoću alata Performance monitor. Također, naučili ste da je sasvim normalno da se OS u cijelosti, ili neki od procesa naizmjence prebacuje između ta dva načina rada. Pogledajmo kako to izgleda.

4.1 Performance Monitor

1. Otvorite **Performance Monitor**
2. Kliknite na **Performance Monitor** opciju iz lijevog okna.
3. Kliknite na gumb **Add** (zeleni plus).
4. Iz grupe **Processor** odaberite **% Privileged Time** te kliknite na gumb **Add** a potom i na **OK.**
5. (OPCIONALNO) Iz donjeg okna isključite prikaz **% Processor Time** brojača.
6. Otvorite nekoliko aplikacija (npr. Notepad, Paint…) a zatim rapidno pomičite miša po ekranu virtualnog računala. Primijetite povećanje vremena brojača u kernel modu. Ovo ilustrira vrijeme koje OS provede obrađujući prekide od strane miša i diskovnog sustava. Pomaci miša se prikazuju unutar GUI-ja. Dotični se, kao što ste već naučili, izvršava u kernel modu pod Windows operacijskim sustavima.
7. Zatvorite Performance Monitor.

----------NAPOMENA----------

Kernel vremena je moguće promatrati i iz **Task Managera**. Jednostavno na kartici Performance desnim klikom miša iznad grafa odaberite **Show Kernel Times**. Na grafovima se prikazuje dodatna crvena linija koja ilustrira vrijeme provedeno u kernel modu. Kao i kod Performance Monitora, ta linija će rasti uslijed radnji kao što je rapidno pomicanje miša

5 Kernel Debugging

Kernel debugging je postupak analize strukture kernela, odnosno, njegovih funkcija. Ovim postupkom se može dobiti uvid u interne informacije Windows OS-a. Te informacije nisu lako dobavljive – potrebno je koristiti specijalizirane alate kao bi dobili jasniju sliku samom kodu Windowsa. Neki od alata koji omogućuju kernel debugging su **Microsoft Windows Debugging Tools i LiveKD** iz Sysinternals grupe alata. Mi ćemo se koncentrirati na LiveKD alat jer ne zahtijeva boot sustava u debugging modu. Debugging mod je opcija koja se odabire na start-up meniju Windowsa (prije boota) i namijenjena je naprednoj analizi grešaka, uglavnom povezanim s upravljačkim programima. Iako bez te opcije nije moguće dobiti apsolutne sve informacije, one za naša razmatranja nisu ni potrebna.

5.1 Ntoskrnl i HAL

Podsjetimo se, **ntoskrnl.exe** datoteka sadrži sam kernel i njegove izvršne servise. Naspram prethodnih verzija Windowsa, Windowsi 7 imaju unificirani kernel neovisno o tome da li se radi o jednoprocesorskom (ili jezgrenom) sustavu, ili višeprocesorskom. Ova promjena je ujedno pokazatelj trenda; naime, današnja PC računala se isporučuju s višejezgrenim procesorima te nema potrebe za različitim verzijama kernela. U doba Windowsa XP je to bilo nužno. Ipak, spomenimo svojevrsni izuzetak. Verzioniranje kernela pod Windowsima 7 je moguće ukoliko imate instaliran 32-bitni OS. 32-bitni OS-ovi mogu koristiti PAE mod (eng. Physical Adress Extension), ukoliko postoji hardverska podrška u obliku **NX** (AMD) ili **XD** (Intel) instrukcija. Na 64 bitnim OS-ovima se PAE ne koristi jer za njim nema ni potrebe. Teoretski, PAE omogućuje adresiranje do 64 GB memorije na 32 bitnim strojevima. Praktično, pod Windowsima 7 ta je vrijednost puno manja i iznosi 4 GB ukupne fizičke memorije. Microsoft ne dopušta PAE mod na klijentskim 32-bitnim OS-ovima. Slična situacija je i s HAL-om: također postoje različite verzije, ovisno o broju izvršnih jezgri. Ipak, na 64-bitnim strojevima postoji samo jedna verzija HAL-a (hal.dll). Naime, 64-bitni sustavi se tretiraju kao da imaju istu matičnu ploču iz razloga što procesor zahtijeva ACPI i APIC podršku za rad. U donjoj tablici možete vidjeti verzije HAL-a.

HAL datoteka	Podržani sustavi
Halacpi.dll	ACPI PC računala. Implicira jednoprocesorski, tj. jednojezgreni sustav bez naprednog kontrolera prekida (APIC).
Halmacpi.dll	APIC (eng. Advanced Programmable Interrupt Controller) PC računalo s ACPI podrškom. Prisustvo APIC-a implicira podršku za simetrično multiprocesiranje.

U donjoj tablici možete vidjeti verzije kernela.

Vrsta datoteke	PAE verzija	Ne-PAE verzija
kernel	Ntkrpamp.exe	Ntkrnlmp.exe
HAL	Ovisno o sustavu	Ovisno o sustavu

5.2 Kako provjeriti verzije kernela i HAL-a?

Iako sami Windowsi ne bilježe na korisniku smislen način podatak o učitanoj verziji HAL-a i kernela (postoji način putem Event Viewera i ključeva u Registryju), pomoću LiveKD alata je tu informaciju moguće vrlo lako dobiti:

1. Pokrenite **Command Prompt** kao administrator (**vidi Dodatak1!**).
2. Podesimo putanju tako da aplikacije znaju mjesto Debuging alata. Upišite: **set path=%path%;C:\Program Files (x86)\Windows Kits\10\Debuggers\x64**
3. Preuzmite i raspakirajte LiveKD s linka: https://docs.microsoft.com/en-us/sysinternals/downloads/livekd
4. Pokrenite alat **livekd64.exe** kao administrator.
5. Upišite naredbu **lm vm nt** (prikaz informacija o kernelu)
6. Provjerite s gornjom tablicom da li radite na PAE omogućenom stroju.
7. Upišite naredbu **lm vm hal** (prikaz informacija o kernelu).
8. Upišite **q** za napuštanje debugera. Kada vas pita da li želite ponovo pokrenuti izaberite **n**.
9. Zatvorite Command Prompt.

5.3 Ovisnost HAL-a i kernela

Odnos između HAL-a i kernela se može provjeriti pomoću alata Dependency Walker (zaseban, besplatni alat):

1. Uključite prikaz skrivenih i sistemskih datoteka u Windows Exploreru.
2. Preuzmite **Dependency Walker** za svoju operacijski sustav sa linka https://www.dependencywalker.com/
3. Kao i kod Proces Monitora nije potrebna instalacija već samo raspakirate komprimiranu datoteku i pokrenete alat kao administrator.
4. Kliknite na gumb **Open** i pozicionirajte se u direktorij Windows\System32
5. Odaberite datoteku **ntoskrnl.exe**

Primijetite da su ntoskrnl i HAL međusobno povezani. Ta povezanost je neraskidiva – oba modula koriste funkcije iz onog drugog. Ntoskrnl je također povezan i sa sljedećim komponentama:

- **Pshed.dll** (eng. Platform-Specific Hardware Error Driver): komponenta čija je uloga glumiti svojevrsni „HAL“ samo za prikaz pogrešaka. Kao i kod pravog HAL-a, namjena je apstrahirati mehanizam prijave grešaka ostatku OS-a.
- **Bootvid.dll** (eng. Boot Video Driver): modul koji omogućuje podršku za VGA grafiku koja se prikazuje kod pokretanja sustava (npr. boot logo Windowsa). Na 64 bitnim sustavima, ovaj modul je ugrađen direktno u kernel radi izbjegavanja konflikata.

- **Kdcom.dll** (eng. Kernel Debugger Protocol): biblioteka s komunikacijskim funkcijama koje se koriste kod boota u debugging modu.
- **Ci.dll** (eng. Code integrity): modul koji provjerava integritet (bolje rečeno, vjerodostojnost) aplikacija, DLL modula ili upravljačkih programa pregledom njihovih digitalnih certifikata.

Dependency Walker se može iskoristiti i za prikaz ovisnosti aplikacija ovisno o njihovoj vrsti (GUI ili linijske):

1. Iz direktorija **\Windows\System32** otvorite datoteku **notepad.exe**.
2. Proučite module s kojima je povezana.
3. Ponovite postupak za datoteku **cmd.exe** i uočite razlike.
4. Zatvorite Dependency Walker.

6 Pristup API-ju

Za kraj današnje vježbe podsjetit ćemo se i API-ja. **Windows API** (eng. Application Programming Interface) je biblioteka s ogromnim brojem funkcija koje programeri pozivaju kako bi svojim aplikacijama omogućile pristup hardveru i servisima. API predstavlja prvi sloj apstrakcije na putu prema „dnu“ računala, čime se znatno olakšava pisanje nativnih WIN32 aplikacija.

Pristup API-ju ćemo demonstrirati pozivom WMI klase **Win32_OperatingSystem** i iščitavanjem njenih atributa. Provjerit ćemo da li virtualno računalo koje koristimo spada u tzv. **provjerenu verziju** (eng. **checked build**). To je specijalna verzija namijenjena debugiranju Windowsa, a dostupna je MSDN i Technet pretplatnicima.

Koriste ju programeri upravljačkih programa iz razloga što ima vrlo detaljne mehanizme provjere grešaka unutar funkcija kernel moda koje upravljački programi pozivaju. Primjerice, ukoliko upravljački program neispravno pozove kernel funkciju, Windowsi će prekinuti s izvršavanjem koda i prikazati informacije o grešci.

Kod „običnih“ verzija Windowsa vjerojatnost da će doći do kritične greške korupcijom sadržaja memorije je, u spomenutom slučaju, vrlo velika. Greška će se manifestirati „popularnim“ **BSOD** (eng. **Blue Screen Of Death**) ekranom i sustav će se morati ponovno pokrenuti. Od cijele debug verzije Windowsa nas će zanimati samo zastavica (eng. Flag) **DBG**, tj. njena vrijednost. Ukoliko radite na debug verziji vrijednost DBG zastavice će biti **1**, dok je u protivnom njena vrijednost **0**.

----------NAPOMENA----------

Donji kod je pisan u **VBS** (Visual Basic Script) jeziku i moguće ga je izvršiti direktno pod Windowsima, bez dodatnih aplikacija ili kompajlera. Komponenta Windowsa koja izvršava takve skripte se zove **Windows Scripting Host**.

1. Otvorite Notepad.
2. Upišite donji kod u Notepad (možete koristiti i Copy/Paste ali morate dodati rasporediti kod u retke kao što je u ovom primjeru):

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
&    "{impersonationLevel=impersonate}!\\"    &    strComputer    &
"\root\cimv2")
Set colOperatingSystems = objWMIService.ExecQuery _
("SELECT * FROM Win32_OperatingSystem")
For Each objOperatingSystem in colOperatingSystems
Wscript.Echo "Opis: " & objOperatingSystem.Caption
Wscript.Echo "Debug: " & objOperatingSystem.Debug
Wscript.Echo "Verzija: " & objOperatingSystem.Version
Next
```

3. Spremite kod u datoteku pod imenom **skripta.vbs** (pazite kod odabira formata datoteke u Notepadu!) na radnu površinu (Desktop).
4. Pokrenite datoteku i zaključite da li radite na debug verziji Windowsa.

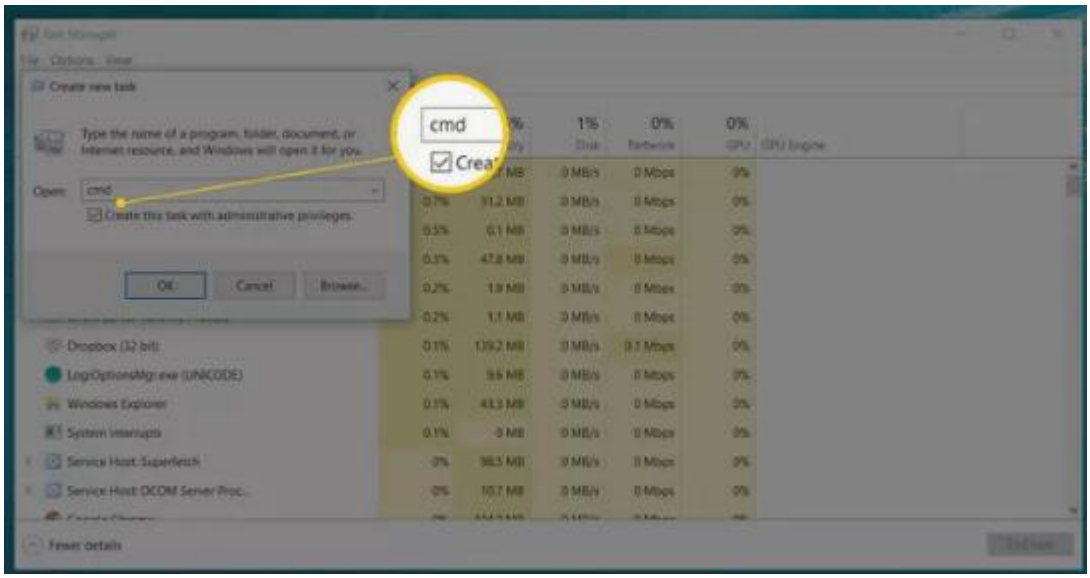
7 Što treba znati nakon ove vježbe?

1. Objasniti ulogu **PID**-a.
2. Objasniti **hijerarhiju** procesa pod Windows OS-om.
3. Objasniti namjenu i ograničenja PAE tehnologije.
4. Opisati alate korištene u vježbi

8 Dodatak 1: Pokretanje CDM (Command Prompt) u administratorskom modu

Za pokretanje CDM alata koji želi pristup sistemskim programima ili servisima moramo raditi sa administratorskim pravima („*admin mod*“), jer je Operativni sustav građen u ljuskama (ring-ovima).

1. Otvorite Task Manager (**CTRL+SHIFT+ESC**)
2. U Task Manageru odaberite: **FILE > Run new task**
   Ukoliko ne vidite „FILE“ odaberite „*More details*“
3. U CREATE NEW TASK upišite „**cmd**“ i označite opciju „**Create this task with administrative privileges**“



4. Pritisnite **OK.** Task Manager možete poslije slobodno ugasiti.