

## Ukeoppgaver 18 – Internminne og powershell

**RØD** - Obligoppgaver

**GUL** – Ikke obligoppgaver

**TURKIS** – Ukens nøtt og utfordringer

Teorioppgaver besvares ved hjelp av hjelpemidler.

### 1. **(Oblig)**

En fysisk adresse er en unik identifikator som tilordnes til hver maskinvarekomponent i en datamaskin, som for eksempel i RAM, harddisk og nettverkskort. Denne adressen brukes av maskinvaren til å identifisere og kommunisere med hverandre direkte i datamaskinen.

En virtuell adresse, derimot, er en adresse som brukes av operativsystemet for å kommunisere med minnet. Denne adressen oversettes til en fysisk adresse av en komponent som kalles MMU. Dette gjør at operativsystemet kan gi inntrykk av at det er mer minne tilgjengelig enn det faktisk er, ved å laste inn deler av programvaren fra harddisken til minnet når det trengs, og så bytte ut deler av minnet når det ikke lenger er nødvendig.

Kort sagt er den fysiske adressen en maskinvareadresse, mens den virtuelle adressen er en adresse som brukes av operativsystemet for å organisere minnet og kommunisere med maskinvaren.

### 2. **(Ikke oblig)**

### 3. **(Oblig)**

Remarkable:

⑤ a) 500 | 2  
250 | 0  
125 | 0  
62 | 1  
31 | 0  
15 | 1  
7 | 1  
3 | 1  
1 | 1  
0 | 1

0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 = 500

0 1 0 0 0 0 1 1 1 1 0 1 0 0

0 1 0 0 0 0 1 1 1 1 0 1 0 0

1016 2032 512 256 128 64 32 16 8 4 2 1

8692 fysisk adresse

går fra 16 bit til 15 bit.

⑥ 18010

18010 | 2  
9005 | 0  
4502 | 1  
2251 | 0  
1125 | 1  
562 | 1  
281 | 0  
140 | 1  
70 | 0  
35 | 0  
17 | 1  
8 | 1  
4 | 0  
2 | 0  
1 | 0  
0 | 1

1000 1100 1011 010

01000 1100101010

10 001100 1011010

16 384  
1024  
512  
64  
26

18010 virtuelle adresse

⑦ 46512

46512 | 2  
23256 | 0  
11628 | 0  
5814 | 0  
2907 | 0  
1453 | 1  
726 | 1  
363 | 0  
181 | 1  
90 | 1  
45 | 0  
22 | 1  
11 | 0  
5 | 1  
2 | 1  
1 | 0  
0 | 1

10 11010110110000

16 384  
16 384  
3 2768

111010110110000 er 30128

#### 4. (Oblig)

Ja, det er mulig at CPU-cachen vil involveres når en x86-instruksjon som henter en byte fra RAM og legger den i et register utføres. Når CPU-en utfører instruksjoner, vil den først sjekke om dataene som instruksjonene trenger allerede finnes i CPU-cachen. Hvis dataene ikke finnes i cachen, vil CPU-en måtte hente dem fra hovedminnet.

#### 5. (Oblig)

Nei, CPU-cache vil ikke involveres når en instruksjon som sammenlikner to tall i registre utføres fordi det ikke krever datainnhenting fra hovedminnet(RAM).

**6. (Oblig)**

En 10 bit-adresse kan adressere  $2^{10} = 1024$  forskjellige minneområder. Siden hvert minneområde representerer en byte, kan det virtuelle adresserommet som kan adresseres med disse 10-bit adressene være 1024 stort.

**7. (Oblig)**

Det totale virtuelle adresserommet for en prosess avhenger av antall biter som brukes til å adressere det, og dette kan variere mellom forskjellige systemer og arkitekturer.

Hvis vi antar at det virtuelle adresserommet er 32-bit, betyr det at det er  $2^{32}$  adresser som hver kan peke til en side på 128 byte. Formelen blir:

$$\frac{2^{32} \text{ bytes}}{128 \text{ bytes}} = 33,554,432 \text{ sider}$$

**8. (Ikke oblig)**

Uttrykket for worst-case mengde disk-plass som trengs for swap-området kan utledes ved å ta hensyn til to tilfeller der all tilgjengelig RAM er i bruk: når alle prosesser bruker sin maksimale tillatte minneallokering, og når alle tilgjengelige virtuelle adresser er i bruk.

La oss anta at swap-området er gitt i størrelse  $s$  bytes. Da vil det maksimale antall prosesser som kan kjøre samtidig, være begrenset av størrelsen på RAM og størrelsen på swap-området, og det vil være gitt av:

$$N = (R + S) / P$$

Der  $P$  er størrelsen på hver prosess' minneallokering.

På samme måte vil det maksimale antallet virtuelle adresser som brukes, være begrenset av størrelsen på RAM og swap-området, og vil være gitt av

$$V = R + S$$

Totalt blir det med de to likningene over:

$$S = V - (N * P)$$

Dette er teoretisk øvrig grense og vil i virkeligheten kreve andre faktorer inkludert de faktiske minnebehovene til prosessene som kjører, og ressursene tilgjengelig på systemet, noe som gjør det urealistisk.

**9. (Ikke oblig)**

10. (Ikke oblig)

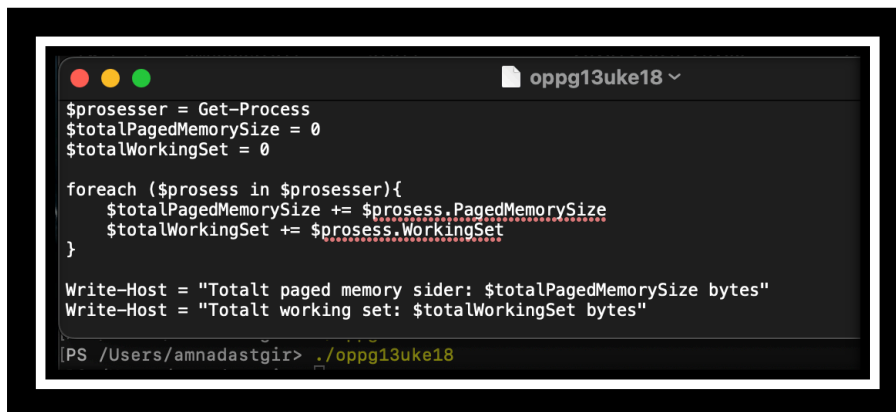
11. (Ikke oblig)

12. (Ikke oblig)

Kan hende det er unødvendig i systemer som er veldig enkle, eller der det ikke er behov for å kjøre applikasjoner samtidig. Det øker unødvendig kompleksitet.

13. (Oblig)

Under er det skript og kjøring. Men jeg får ikke kjørt, teksteditoringsverktøyet bare åpnes opp.



```
$prosesser = Get-Process
$totalPagedMemorySize = 0
$totalWorkingSet = 0

foreach ($prosess in $prosesser){
    $totalPagedMemorySize += $prosess.PagedMemorySize
    $totalWorkingSet += $prosess.WorkingSet
}

Write-Host "Totalt paged memory sider: $totalPagedMemorySize bytes"
Write-Host "Totalt working set: $totalWorkingSet bytes"

[PS /Users/amnadastgir> ./oppg13uke18
```

14. (Oblig)

Med andre ord er WS størrelsen på det settet med minnesider som en prosess har tilgang til i RAM.



**Working set (Windows)** = det sett av sider som en prosess har brukt nylig. Samme som RES i Linux.

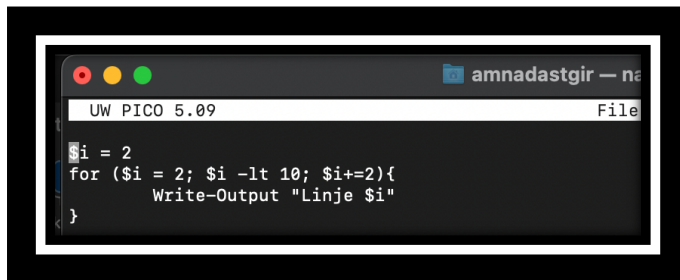
På en Windows med ca 360 MByte internminne kan det være et problem hvis de totale summene av WS og PM for alle prosessene som kjører, er større enn den totalte mengden RAM. Hvis dette skjer, vil systemet tvinges til å paginere minne ut til disk (i pagefile) for å frigjøre plass i RAM. Dette vil føre til at systemytelsen reduseres betraktelig, siden å lese og skrive fra disk er mye tregere enn å lese og skrive fra RAM.

I eksempelet i oppgaven er PM større enn WS. Dette betyr at del av PM allerede er plassert på disken (paged out), mens WS er den faktiske mengden av minne som brukes av prosessen for øyeblikket.

Hvis WS + PM for alle prosessene som kjører overstiger mengden av tilgjengelige RAM, kan systemet bli tregt og ytelsen påvirkes negativt. I dette tilfellet ville det være lurt å undersøke hva som forbruker så mye minne og eventuelt optimere eller begrense ressursbruken for å forbedre systemytelsen.

### 15. (Oblig)

Mitt PowerShell skript:



```
UW PICO 5.09 File
i = 2
for ($i = 2; $i -lt 10; $i+=2){
    Write-Output "Linje $i"
}
```

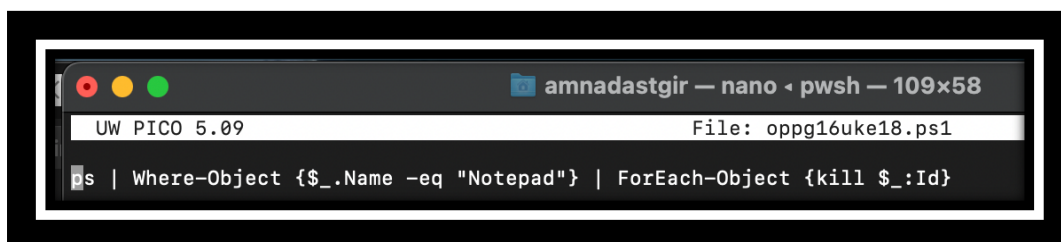
Kjøringen:



```
[PS /Users/amnadastgir> ./oppg15uke18.ps1
Linje 2
Linje 4
Linje 6
Linje 8
```

### 16. (Oblig)

Skriptet mitt som sletter alle prosesser dersom notepad eksisterer. Men det går ikke an å åpne notepad så langt som jeg vet.



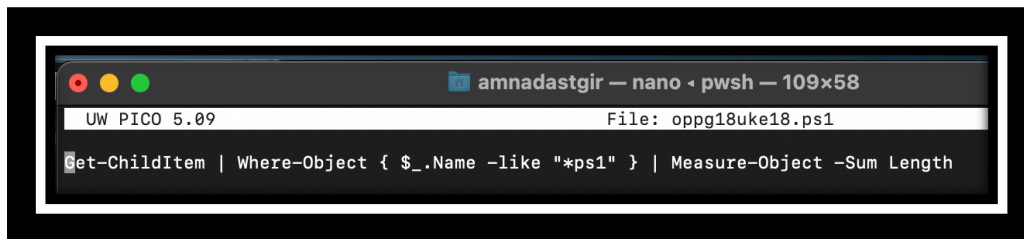
```
UW PICO 5.09 File: oppg16uke18.ps1
ps | Where-Object {$_.Name -eq "Notepad"} | ForEach-Object {kill $_.Id}
```

### 17. (Ikke oblig)

Oversette kode fra c til PowerShell, men jeg gidder ikke.

### 18. (Oblig)

Under er koden min:



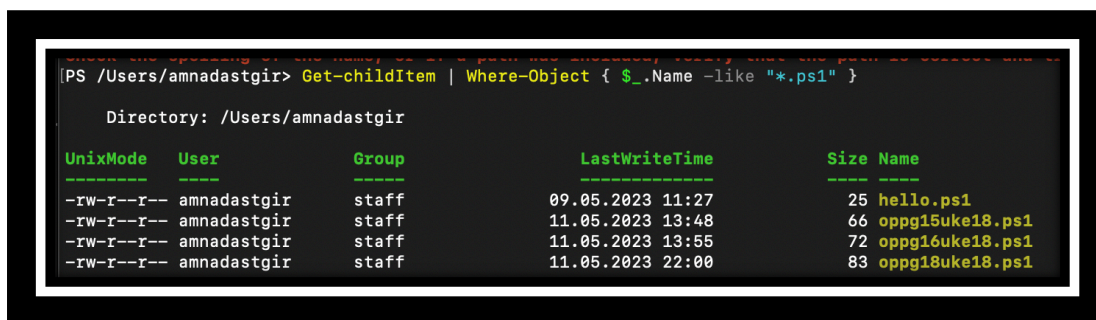
```
amnadastgir — nano · pwsh — 109x58
UW PICO 5.09 File: oppg18uke18.ps1
Get-ChildItem | Where-Object { $_.Name -like "*.ps1" } | Measure-Object -Sum Length
```

Under er summen fra kjøring av skriptet:



```
PS /Users/amnadastgir> ./oppg18uke18.ps1
Count           : 4
Average          : 
Sum              : 246
Maximum          : 
Minimum          : 
StandardDeviation : 
Property         : Length
```

Under listet jeg opp alle filene som faktisk ender med .ps1 for å finne regne om skriptet er riktig.



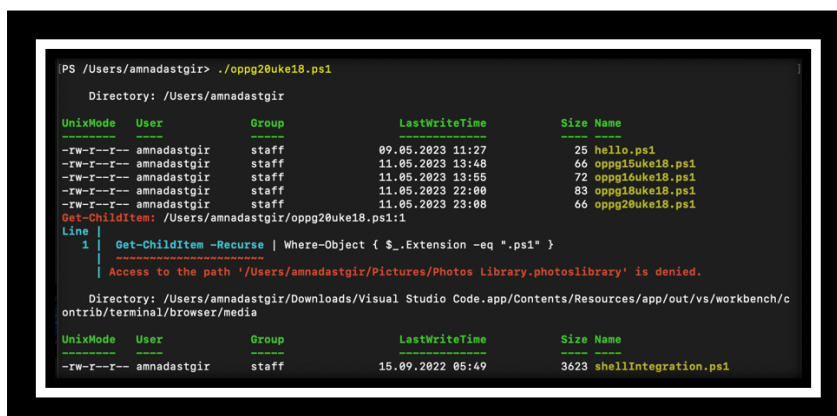
```
PS /Users/amnadastgir> Get-ChildItem | Where-Object { $_.Name -like "*.ps1" }

Directory: /Users/amnadastgir

UnixMode User      Group      LastWriteTime         Size Name
-----
-rw-r--r-- amnadastgir staff      09.05.2023 11:27         25 hello.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 13:48         66 oppg15uke18.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 13:55         72 oppg16uke18.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 22:00         83 oppg18uke18.ps1
```

19. (Ikke oblig)

20. (Oblig)



```
PS /Users/amnadastgir> ./oppg20uke18.ps1

Directory: /Users/amnadastgir

UnixMode User      Group      LastWriteTime         Size Name
-----
-rw-r--r-- amnadastgir staff      09.05.2023 11:27         25 hello.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 13:48         66 oppg15uke18.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 13:55         72 oppg16uke18.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 22:00         83 oppg18uke18.ps1
-rw-r--r-- amnadastgir staff      11.05.2023 23:08         66 oppg20uke18.ps1

Get-ChildItem: /Users/amnadastgir/oppg20uke18.ps1:1
Line |
1 | Get-ChildItem -Recurse | Where-Object { $_.Extension -eq ".ps1" }
    ~~~~~
    Access to the path 'C:\Users\amnadastgir\Pictures\Photos Library.photoslibrary' is denied.

Directory: /Users/amnadastgir/Downloads/Visual Studio Code.app/Contents/Resources/app/out/vs/workbench/c
ontrib/terminal/browser/media

UnixMode User      Group      LastWriteTime         Size Name
-----
-rw-r--r-- amnadastgir staff      15.09.2022 05:49        3623 shellIntegration.ps1
```

Skriptet mitt under:

```

    amnadastgir — nano < pwsh — 109x58
    UW PICO 5.09                               File: oppg20uke18.ps1
    Get-ChildItem -Recurse | Where-Object { $_.Extension -eq ".ps1" }
  
```

21. (Ikke oblig)

22. (Oblig)

Amna D
Operativsystemer
21.04.23

Vi fikk lov til å kjøre filen det betyr at 'Get-ExecutionPolicy' den henter ut policy. Under kan vi se at Hårek allerede har satt den til 'RemoteSigned':

```

PS C:\Users\os> Get-ExecutionPolicy
RemoteSigned
  
```

Default er den ellers satt til restricted. Under kan vi se på bildet at håret ikke får tillatelse til å endre det tilbake til 'restricted'. Dette er fordi Hårek har en vanlig bruker, for å få lov til dette her så må han starte PowerShell som en administrator.

```

PS C:\Users\os> Set-ExecutionPolicy Default
Set-ExecutionPolicy : Access to the registry key 'HKLM\LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell\PowerShell' is denied. To change the execution policy for the default (localMachine) scope, start with the "Run as administrator" option. To change the execution policy for the current user, run "Set-ExecutionPolicy"
Set-ExecutionPolicy : Set-ExecutionPolicy Default
+ ~~~~~
+ CategoryInfo          : PermissionDenied; ([]) [Set-ExecutionPolicy], UnauthorizedAccessException
+ FullyQualifiedErrorId : System.UnauthorizedAccessException,Microsoft.PowerShell.Commands.Set-Exe
  
```

Da kan man ved å taste Windows + r og PowerShell, da får man opp et vanlig PowerShell vindu.

Se på bildet til Høyre og høyreklikk på Windows PowerShell da må du velge kjør som administrator for å åpne et PowerShell som administrator.

Da vil det være mulig å sette execution policy til default.

```

PS C:\Windows\System32> Set-ExecutionPolicy default
Execution Policy change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_execution_policies help topic at https://go.microsoft.com/fwlink/?LinkId=11379. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"):
  
```

Når Hårek går tilbake til det vanlige shellet så kan han se at han ikke får lov til å kjøre skriptet, lenger og dette er fordi han må ha satt execution policy tilbake til default. Hvis han blir endre dette og kjører skriptet må han tilbake til administrator shellet og endre execution-policy til set-ExecutionPolicy RemoteSigned.

23. (Oblig)

Kjøring under:

```
PS /Users/amnadastgir> ./oppg23uke18.ps1
Get-ItemProperty: /Users/amnadastgir/oppg23uke18.ps1:1
Line |
1 | Get-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\PowerShell\1\ShellId ...
      | ~~~~~
      | Cannot find drive. A drive with the name 'HKLM' does not exist.
```

Skriptet mitt:

```
amnadastgir — nano • pwsh — 109x58
UW PICO 5.09 File: oppg23uke18.ps1
Get-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell' |
Select-Object ExecutionPolicy
```

#### 24. (Oblig)

Jeg får ikke kjørt og vet derfor ikke helt om skriptet er riktig men her er koden:

```
amnadastgir — nano • pwsh — 109x58
UW PICO 5.09 File: oppg24uke18.ps1
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell" -Name
"ExecutionPolicy" -Value "RemoteSigned"
```

#### 25. (Ikke oblig)

#### 26. (Ikke oblig)

#### 27. (Ikke oblig)

#### 28. (Oblig)

Under er koden min:

```
amnadastgir — nano • pwsh — 109x58
UW PICO 5.09 File: oppg28uke18.ps1
$dato = Get-Date -Year 1814 -Month 5 -Day 17
$ukedag = $dato.DayOfWeek
Write-Output "Ukedag for 17 mai 1814 var: $ukedag"
```

Kjøring og resultat:



```
PS /Users/amnadastgir> ./oppg28uke18.ps1
Ukedag for 17 mai 1814 var: Tuesday
```

29. (Ikke oblig)

30. (Ikke oblig)

31. (Ikke oblig)

### 32. UKENS NØTT 1

Inne i funksjonen skal A og B adderes. Den første metoden viser at parenteser bare skriver ut tallene tilbake. Hvis man egentlig skal sende inn tall og vil at det skal komme ut som en sum, må man bare skrive funksjonsnavnet og tallene ved siden av.

```
PS C:\Users\os> function add($a,$b) { $a + $b }
PS C:\Users\os> add(2,3)
2
3
PS C:\Users\os> add 2 3
5
```

### 33. UKENS NØTT 2

ork