

Tasks on Shell Scripting

1. use the 'bc' command to perform all arithmetic operations

```
vineet@DESKTOP-M1MSH4M: /mnt/c/Users/Vineet
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ read num1
10
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ read num2
20
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ echo "$num1 + $num2" | bc
30
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$
```

2. Write a shell script to find the total disk space and the space Utilized.

```
vineet@DESKTOP-M1MSH4M: /mnt/c/Users/Vineet
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          248G  95G  153G  39% /
none            248G  95G  153G  39% /dev
none            248G  95G  153G  39% /run
none            248G  95G  153G  39% /run/lock
none            248G  95G  153G  39% /run/shm
none            248G  95G  153G  39% /run/user
tmpfs           248G  95G  153G  39% /sys/fs/cgroup
C:\             248G  95G  153G  39% /mnt/c
E:\             293G 120G  174G  41% /mnt/e
F:\             391G 205G  186G  53% /mnt/f
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$
```

3. Write a shell script to check if the host (server) is active and client is connected if not display an message that could not connect to the server.

```
vineet@DESKTOP-M1MSH4M: /mnt/c/Users/Vineet
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ ping -c1 -W1 $ip_addr && echo 'server is up ' || echo 'server is down'
PING google.com(dell11s22-in-x0e.1e100.net (2404:6800:4002:82d::200e)) 56 data bytes
64 bytes from dell11s22-in-x0e.1e100.net (2404:6800:4002:82d::200e): icmp_seq=1 ttl=119 time=20.6 ms

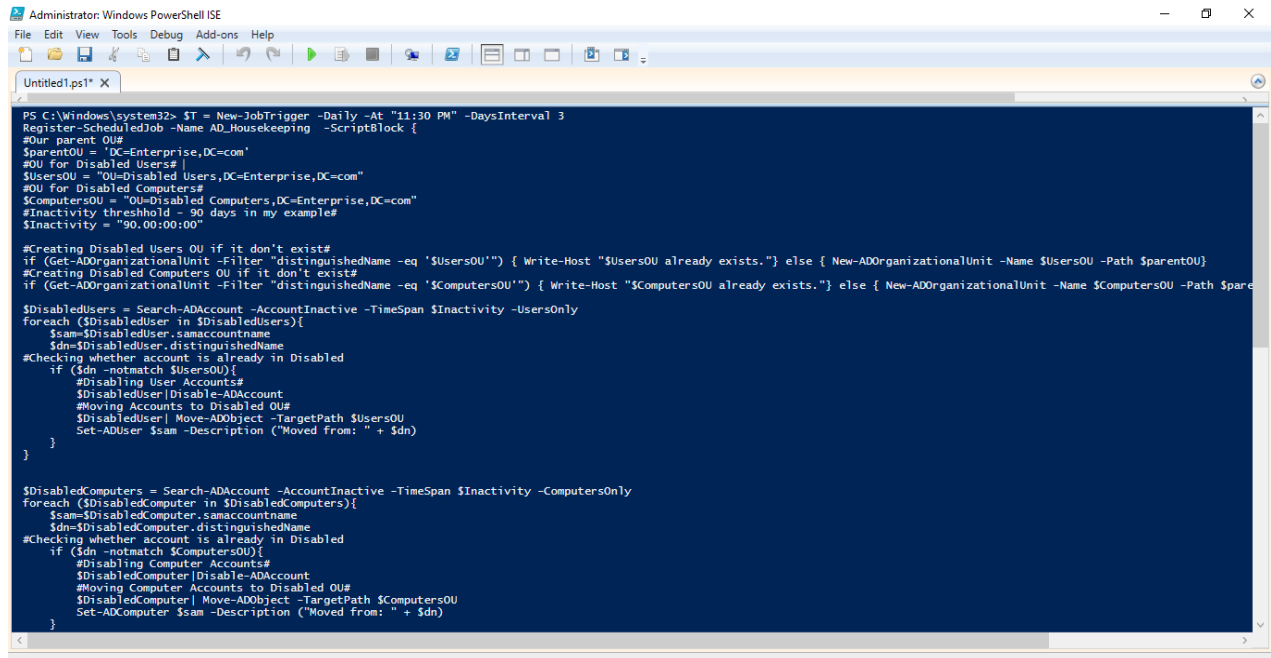
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 20.681/20.681/20.681/0.000 ms
server is up
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$
```

4. Write a shell script to delete all the old files in /home directory.
if the file is more than 6 months old then delete the file.
hint :: use find and touch commands.

```
vineet@DESKTOP-M1MSH4M: /mnt/c/Users/Vineet
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ find vimfiles type f -mtime +30
vimfiles/colors
vimfiles/compiler
vimfiles/doc
vimfiles/ftdetect
vimfiles/ftplugin
vimfiles/indent
vimfiles/keymap
vimfiles/plugin
vimfiles/syntax
find: 'type': No such file or directory
find: 'f': No such file or directory
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$ find vimfiles -name "*.log" -type f -mtime +30 -delete
vineet@DESKTOP-M1MSH4M:/mnt/c/Users/Vineet$
```

5. Write a shell script to create the back up of all the files that are present in the machine.

6. Write a shell script to disable all the inactive users.



```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1 X
PS C:\Windows\system32> $T = New-JobTrigger -Daily -At "11:30 PM" -DaysInterval 3
Register-ScheduledJob -Name AD_Housekeeping -ScriptBlock {
#Our parent OU#
$parentOU = 'DC=Enterprise,DC=com'
#OU for Disabled Users#
$UsersOU = "OU=Disabled Users,DC=Enterprise,DC=com"
#OU for Disabled Computers#
$ComputersOU = "OU=Disabled Computers,DC=Enterprise,DC=com"
#Inactivity threshold = 90 days in my example#
$Inactivity = "90.00:00:00"

#Creating Disabled Users OU if it don't exist#
if (Get-ADOrganizationalUnit -Filter "distinguishedName -eq '$UsersOU'" ) { Write-Host "$UsersOU already exists." } else { New-ADOrganizationalUnit -Name $UsersOU -Path $parentOU}
#Creating Disabled Computers OU if it don't exist#
if (Get-ADOrganizationalUnit -Filter "distinguishedName -eq '$ComputersOU'" ) { Write-Host "$ComputersOU already exists." } else { New-ADOrganizationalUnit -Name $ComputersOU -Path $parentOU}

$DisabledUsers = Search-ADAccount -AccountInactive -TimeSpan $Inactivity -UsersOnly
foreach ($DisabledUser in $DisabledUsers){
    $sam=$DisabledUser.samaccountname
    $dn=$DisabledUser.distinguishedName
    #Checking whether account is already in Disabled
    if ($dn -notmatch $UsersOU){
        #Disabling User Accounts#
        $DisabledUser | Disable-ADAccount
        #Moving Accounts to Disabled OU#
        $DisabledUser | Move-ADObject -TargetPath $UsersOU
        Set-ADUser $sam -Description ("Moved from: " + $dn)
    }
}

$DisabledComputers = Search-ADAccount -AccountInactive -TimeSpan $Inactivity -ComputersOnly
foreach ($DisabledComputer in $DisabledComputers){
    $sam=$DisabledComputer.samaccountname
    $dn=$DisabledComputer.distinguishedName
    #Checking whether account is already in Disabled
    if ($dn -notmatch $ComputersOU){
        #Disabling Computer Accounts#
        $DisabledComputer | Disable-ADAccount
        #Moving Computer Accounts to Disabled OU#
        $DisabledComputer | Move-ADObject -TargetPath $ComputersOU
        Set-ADComputer $sam -Description ("Moved from: " + $dn)
    }
}
```