```powershell
#Remove-OldHomeDirectory.ps1
#Written by: Jeff Brusoe
#Last Updated: April 15, 2020

[CmdletBinding()]
param (
    [string]$FileSystemPath = "\\hs.wvu-ad.wvu.edu\Public\Path", #Chosen for test purposes,
    [string]$ADOrgUnitCN = "OU=path,OU=SOM,OU=HSC,DC=hs,DC=wvu-ad,DC=wvu,DC=edu"
    )

$Error.Clear()
Set-StrictMode -Version Latest
Clear-Host

Function Exit-Command([string]$ExitMessage)
{
    Write-Warning "$ExitMessage`nProgram is exiting."
    Stop-Transcript
    return
}

$SessionTranscript = "$PSScriptRoot\Logs\" + (Get-Date -format yyyy-MM-dd-HH-mm) + "-SessionTranscript.txt"
Start-Transcript $SessionTranscript

#Verify file system path exists
if ([string]::IsNullOrEmpty($FileSystemPath))
{
    Exit-Command "Path not specified."
}

if (Test-Path $FileSystemPath)
{
    Write-Output "File system path exists."
}
else
{
    Exit-Command "File system path does not exist."
}

#Create directory not found file
$OUName = $FileSystemPath.SubString($FileSystemPath.LastIndexOf("\")+1)
Write-Output "OU Name: $OUName"

$NotFoundFile = "$PSScriptRoot\Logs\" + (Get-Date -format yyyy-MM-dd-HH-mm) + "-$OUName-NotFoundFile.txt"
New-Item -type file -Path $NotFoundFile -Force

Add-Content -Path $NotFoundFile -Value "Search Path: $ADOrgUnitCN"
Add-Content -Path $NotFoundFile -Value "Home directories with no corresponding users:"

try
{
    Write-Output "Generating list of AD users"
    Write-Output "Search Base: $ADOrgUnitCN"
    $users = Get-ADUser -Filter * -SearchBase $ADOrgUnitCN -Properties extensionAttribute1 -ErrorAction Stop
}
catch
{
    Exit-Command "Error getting AD users."
}

if (($users | Measure).Count -gt 0)
{
    $usernames = $users.SAMAccountName
}
else
{
    Exit-Command "No users were found."
}

Write-Output "Username Array:"
Write-Output $usernames

Write-Output "`nAD User Information:"

foreach ($user in $users)
{
    #This is here for logging output purposes
    $SAMAccountName = $user.SAMAccountName

    Write-Output "Current User: $SAMAccountName"
    Write-Output $("End Access Date: " + $user.extensionAttribute1)
    Write-Output $("Enabled: " + $user.Enabled)
    Write-Output "Distinguished Name:"
    Write-Output $user.DistinguishedName

    Write-Output "***************************"
}

$ExcludedDirectories = "shared","ACGME"

$Directories = Get-ChildItem $FileSystemPath -Directory
$DirectoryNames = $Directories.Name

Write-Output $DirectoryNames #Test output

foreach ($Directory in $Directories)
{
    #This is used simply as a output for the transcript
    Write-Output $("Current Directory: " + $Directory.Name)
    Write-Output $("Full path:" + $Directory.FullName)

    #Directory permissions
    #See this link: https://www.petri.com/how-to-get-ntfs-file-permissions-using-powershell
    Write-Output "Directory Permissions:"

    try
    {
        Get-Acl -Path $Directory.FullName -ErrorAction Stop | Format-Table -Wrap
    }
    catch
    {
        Write-Warning "Error retrieving directory permissions."
    }

    if ($usernames -contains $Directory.Name)
    {
        Write-Output "Valid Home Directory"
    }
    else
    {
        Write-Output "Old Home Directory"
        Add-Content -Path $NotFoundFile -Value $Directory.Name
    }

    Write-Output "***************************"
}

Stop-Transcript
```