Universitat de Girona	Enginyeria Informàtica	Juan José Gómez Villegas u1987338
	Assignatura: Sistemes Operatius (P.Inf 1)	

Practica 5 - Treball de Sistemes Operatius: PowerShell

Índex

La meva experiència amb PowerShell

PowerShell m'ha semblat que s'assembla bastant a bash, però més fàcil, per exemple:
Treballar amb variables és més còmode, ja que no cal declarar-les a dalt com es fa en bash, i també he vist que treballa amb objectes, per exemple: fent Get-ChildItem -Path "." per obtenir el contingut del directori actual, el resultat es pot guardar a una variable i per accedir al nom o saber el tipus de fitxer només cal fer $variable.Name o $variable.Mode, la part difícil per saber el tipus està a diferenciar si és un fitxer o un directori, si és un fitxer a Mode hi haurà "-a----", i si hi ha "d-----" llavors és un directori.
També he vist que en PowerShell es poden fer consultes a un objecte o array amb Where-Object (per filtrar) i Select-Object (per seleccionar) com si fos SQL, i es poden fer servir pipes com en bash, això fa més fàcil la part d'obtenir només els directoris de mesures o els fitxers a copiar del directori mesures.
Per sortir d'una funció en cas d'error es fa igual que en bash (instrucció exit), els paràmetres de l'script es passen a través d'un array $args en què el primer paràmetre comença a $args[0] (en bash es passaven en un string i $0 era el nom de l'script).
Per manipular strings PowerShell té les funcions que tindria un llenguatge de programació d'alt nivell, i per fer una expressió regular que validi els paràmetres s'ha de fer servir $variable -match "", i la resta funciona igual que en bash.
Hi ha algunes comandes de bash que canvien de nom, com per fer un echo es fa fent servir la instrucció Write-Output.
I al final, fer la pràctica 1 m'ha semblat més fàcil en PowerShell que en bash, perquè només ha sigut adaptar la pràctica 1 de bash a PowerShell.
L'únic problema que he tingut ha sigut a l'hora d'intentar construir el nom que tindrà el fitxer destí en una funció a part, que no concatenava bé la ruta dels fitxers origen i destí, i en executar la comanda cp donava error de què no trobava la ruta, al final la solució ha sigut construir la ruta d'origen i destí, i copiar els fitxers a dins del bucle foreach.

```
foreach ($j in $fitxers) { # i va iterant per cada fitxer de $i, fent una còpia de $j amb el nom de $i afegit
    $origen = $ruta + "/" + $i.Name + "/" + $j.Name
    $desti = $j.Name.Substring(0, 1) + "_" + $i.Name + ".out"

    cp $origen $desti # còpia el fitxer $j

    $fcopiats = $fcopiats + 1 # i incrementa el nombre de fitxers copiats
}
```

Nota: L'script mostres.ps1 es pot executar des del mateix directori on estigui el directori mesures, o des de dins del directori mesures (igual que amb l'script mostres de bash).