Praktikum 3
Shell Programing

I. Tujuan
1. memahami konsep shell programing
2. memahami jenis-jenis variabel dalam shell
3. mampu menulis program dengan shell programing
4. Memahami konsep Shell interaktif
5. Mengetahui environment Shell

II. Dasar teori
Shell adalah sebuah bahasa penterjemah perintah (command interpreter language) atau sebuah prosesor makro yang menjalankan perintah. Shell juga dapat berarti interpreter pertintah yang menjadi antarmuka antara user dengan utilitas dan bahasa pemrograman. Dengan shell, dapat dibuat sebuah perintah atau file yang berisi perintah-perintah itu sndiri. Perintah baru tersebut mempunyai status yang sama dan di letakkan pada direktori /bin.
shell mengizinkan eksekusi perintah secara synchronously dan asynchronously. Shell menunggu perintah synchronous untuk dilengkapi sebelum menyetujui lebih banyak input, sedangkan perintah asynchronous terus berjalan dalam paralel dengan shell ketika membacanya dan menjalankan perinntah tambahan. Shell juga mengenal adanya redirection. Dengan redirector dapat dilakukan kontrol untuk intput dan output dari perintah yang ada dan juga melakukan kontrol terhadap isinya. Secara default, shell UNIX juga menyediakan perintah-perintah buil-in, seperti pwd, cd, kill, histroy, atau utilitas lain yang terpisah.
Sebagai interface dan command interpreter, shell dapat digunakan secara interaktif maupun noninteraktif. Dangan dua mode tersebut, shell mampu menerima input dari device (keyboard) atau file. Fitur interaktif yang termasuk di dalamnya adalah kontrol job, history, alias, dan editor command line.
Hal lain yang menjadikan shell sangat penting adalah shell menyediakan bahasa pemrograman yang telah disertakan(embeded). Sama halnya dengan bahasa pemrograman tingkat tinggi laennya, interpreter shell juga menyediakan variabel, flow control, quoting dan fungsi.

Macam-macam shell
pada UNIX/Linux terdapat berbagai macam shell dengan kelebihan dan kekurangan masing-masing. Dengan banyaknya variasi shell ini, user bebas memilih shell yang digunakan. Meskipun kebanyakan sistem operasi telah menentukan sebuah shell sebagai shell default, tetapi tidak menutup kemungkinan shell lain juga dapat dijalankan. Berikut ini beberapa macam shell yang umum terdapat dalam sistem operasi UNIX/Linux.

- Bourne Shell (/bin/sh)
- Bourne Again Shell(/bin/bash)
- C Shell (/bin/csh)
- Tenex C Shell (/bin/tcsh)
- Tcl shell (/bin/tclsh)
- Korn shell (shell /bin/ksh)
- Public domain korn shell (/bin/pdksh)
- A shell (/bin/ash)
- Z shell (/bin/zsh)

Mengganti dan menjalankan Shell
Linux menggunakan bash sebagai shell default, tetapi pengguna bisa mengubah shell default untuk tiap usernya. Untuk melihat shell yang sedang digunakan oleh user bisa dilihat pada file

/etc/passwd.
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh

Isi dari file tersebut tiap barisnya dibagi menjadi tujuh bagian, dan setiap terakhir digunakan untuk mendefinisikan shell yang digunakan. Dalam contoh di atas user root menggunakan shell Bash sedangkan daemon menggunakan shell bourne shell.
Cara lain yang bisa digunakan untuk melihat shell adalah dengan melihat environmet user dengan menjalankan perintah env. Environment user merupakan lingkungan user yang berisi semua variabel atau ketentuan khusu untuk user tersebut.

$env
.....
SHELL=/bin/bash
.....

Mengganti Shell
Ada beberapa cara yang dapat digunakan untuk mengubah shell default, yaitu dengan utilitas userconf, atau dapat juga dengan mengubah pada file /etc/passwd tersebut tetapi harus sebagai root.
Contoh mengganti shell dengan utilitas chsh (change shell).

$chsh
Password:
Changing the login shell for praktikum
Enter the new value, or press ENTER for the default
Login Shell [/bin/bash]: /bin/sh

Menjalankan shell
sebuah shell dapat dijalankan tanpa harus mengubah default shell-nya. Cara yang digunakan adalah dengan memanggil nama shell pada command prompt. Misalnya, user akan menggunakan shell sh, user tinggal menjalankan shell sh, user tinggal menjalankan sh sehingga prompt akan berubah sesuai dengan promt Bourne shell.
Untuk keluar dan kembali ke shell default, ketik exit atau tekan Ctrl + d.

bash@praktikum:~$ sh
$
$ exit

Menjalankan script shell
Untuk menjalankan sebuah script shell, sebaiknya lebih dulu memahami penggunaan path absoluth dan path relatif. Ada dua cara yang digunakan untuk menjalankan sebuah shell script, yaitu

$bash hello.sh

./hello.sh

Untuk dapat menjalankan perintah di atas, file program harus dijalankan sebagai file executable. Untuk memberi atribut eksekusi tersebut, dapat digunakan perintah chmod.

$chmod +x hello.

VARIABEL
variabel adalah sebuah kata yang mempunyai nilai. Shell sebagai sebuah interpreter juga menyediakan fasilitas atau kemampuan yang memungkinkan user untuk membuat, mendefinisikan dan menghapus variabel. Sebuah variabel secara umum didefinisikan dengan sintaks berikut :
Nama_variabel=isi variabel

Macam-macam Variabel

Ketika sebuah shell dijalankan, akan ada tiga macam variabel yang secara otomatis dipanggil. Variabel-variabel tersebut adalah :

variabel lokal

variabel lokal adalah variabel yang ada hanya pada saat masih aktif, dan hanya dikenal di lingkungan itu sendiri, sehingga variabel lokal hanya berlaku pada lingkungan dimana variabel tersebut dibuat.

contoh local variabel

```
#!/bin/bash
        HELLO=Hello
        function hello {
            local HELLO=World
            echo $HELLO
        }
        echo $HELLO
        hello
        echo $HELLO
```

Variabel lingkungan

variabel lingkungan adalah variabel yang terdapat dalam shell dan digunakan dalam proses anak yang dijalankan oleh shell tersebut. Variabel lingkungan ini bisa berupa dari variabel lokal yang diekspor. Untuk mengganti variabel lingkungan digunakan perintah export. Contoh berikut adalah mengubah charset lokal menjadi lokal Indonesia.

```
$export LC_LOCAL=id_ID
```

Variabel shell

Variabel shell adalah variabel yang ditetapkan oleh shell dan digunakan oleh shell agar berjalan dengan baik. Sebenarnya, variabel ini bisa dimasukkan dalam kategori variabel lingkungan. Contoh variabel ini adalah variabel defaul dari bash, misalnya:

HOME, PWD, PS1 dan PS2.

Variabel Read-Only

Variabel read-only adalah variabel yang mempunyai atribut read-only, artinya variabel itu tidak bisa diganti nilainya. Bahkan sebuah variabel tidak bisa dihapus dengan perintah unset jika sebuah variabel diberi atribut read-only.

```
$nama=praktikum
$readonly nama
$nama=os
bash:nama:readonly variable
```

Quoting

Quoting adalah mekanisme untuk melindungi metakarakter dari interpretasi sebagai sebuah simbol. Shell juga mempunya beberapa karakter yang difungsikan untuk melindungi metakarkater agar tetap diinterpretasikan sebagai karakter biasa. Ada tiga karakter quoting dalam Shell, yaitu :

1. Backslash (\ )
2. Petik tunggal (')
3. Petik ganda (“)

Contoh quoting dalam Shell :

```
$ echo don\'t miss it
don't miss it
$ echo “don't miss it”
```

don't miss it

Keterangan :
1. Tanda \ menandakan katakter ' yang mengikuti bukan sebuah metakarakter
2. Pengunaan tanda petik double “ juga berfungsi melindungi interpretasi karakter ' sebagai metakarakter

Metakarakter Dalam Shell

Metakarakter adalah sebuah karakter yang memiliki arti tertentu. Dalam Shell juga dikenal beberapa metakarakter. Karena metakarakter juga ada dalam Shell maka yang perlu diperhatikan adalah kesalahan dalam penanganan sebuah karakter. Dalam sebuah kasus mencetak sebuah strin di layar monitor, terkadang terjadi kasus dimana dari salah satu karakter dalam string tersebut merupakan metakarater. Karena mengandung metakarakter maka Shell akan menginterpretasikan string tersebut tidak seperti yang diharapkan.

Contoh kasus :

```
$ echo don't miss it 'enter'
```

Keterangan :
1. Tanda ' dinterpretasikan sebagai serangkaian string sehingga Shell akan menunggu sampai tanda ' berikutnya untuk berhenti dan kemudian menampilkannya.

Jika maksudnya adalah untuk mencetak string #don#t miss it# maka yang perlu diperhatikan adalah

```
$ echo don\'t miss it 'enter'
  don't miss it
```

Keterangan :
1. \ merupakan karakter yang meloloskan interpretasi tanda ' yang merupakan metakarakter dalam Shell.
2. Karakter \ dikenal dengan istilah quoting dalam Shell

Perintah Echo

Echo adalah perintah untuk menampilkan data yang ada pada argumen ke standard output (stdout), yang dalam hal ini stdout bisa merupakan layar monitor atau juga sebuah file.

Perintah Echo dalam Shell memiliki opsi-opsi untuk membentuk atau memberikan format pada data yang dikeluarkan. Sama halnya dengan pemrograman yang lain misalnya bahasa C. karakter yang digunakan untuk membentuk sebuah format dalam perintah echo biasa disebut “escape sequences character”. Contoh escape sequences character adalah \n yang memiliki arti ganti baris atau baris baru. Untuk bisa menggunakan escape sequences dalam Shell yang perlu diperhatikan adalah bahwa secara default shell tidak menerima escape sequence, namun untuk bisa menggunakannya perlu ditambahkan beberapa opsi yang ada dalam perintah Echo. Berikut tabel opsi dan escape sequence dalam perintah Echo.

Fungsi

Fungsi adalah skrip yang berisi kumpulan perintah yang berada diluar program utama. Fungsi biasanya berisi perintah-perintah dalam Shell. Tujuan dari adanya fungsi adalah untuk lebih mengefisiensikan pemangilan sekumpulan perintah yang berulang-ulang pada program yang dibuat.

Di dalam Shell fungsi juga bisa didefinisikan interaktif maupun secara skrip program, dan meskipun didefinisikan secara interaktif, sebuah fungsi juga bisa dipanggil melalui skrip yang dibuat dalam sebuah file dengan catatan fungsi tersebut sudah di export. Setelah melalui mekanisme export ini

sub-shell juga bisa memanggil fungsi tersebut.

Bentuk umum dalam mendefinisikan fungsi dalam BASH Shell adalah sebagai berikut :

```
nama_fungsi () { command; command; }
function nama_fungsi { command; command; }
function nama_fungsi () { command; command; }
```

Array

Pada versi BASH 2.x terdapat fungsi untuk mendefinisikan array satu dimensi. Array memungkinkan seorang programmer mengkoleksi daftar beberapa nilai dalam sebuah variabel. Untuk mengektraksi kembali nilai-nilai tersebut dapat dilakukan dengan menyebutkan nama varibel yang diikuti oleh nomer indek array tersebut.
Pendefinisian sebuah array juga bisa dilakukan on the fly(tanpa mendefinisikan terlebih dahulu). Dan tidak ada batasan maksimum dari sebuah array yang dibuat dalam lingkungan BASH Shell. Pada saat sebuah nilai diberikan ke dalam sebuah array yang telah didefinisikan, indek array secara otomatis akan dimulai dari 0, dan bertambah naik 1 sampai semua kumpulan nilai-nilai dimasukkan.

III. Petunjuk praktikum

1. mulailah menuslis program dengen shell, misal “hello word”.
2. Cobalah sebagai latihan

Latihan 1

```
$ if grep “root” /etc/passwd
> then
> echo “disini ada user yang bernama root”
> else
> echo “user tersebut tidak ada”
> fi
```

Latihan 2

```
#!/bin/bash
echo “selamat datang $USER”
echo “di shell programming”
```

Latihan 3

```
#!/bin/bash
echo “Shell yang digunakan adalah $SHELL”
echo “saat ini jam `date +%T`”
echo “tanggal `date +%D`”
```

Latihan 4

```
#!/bin/bash
echo “hari ini tanggal `date +%d` bulan `date +%m` tahun `date
+%y`”
echo “:D”
```

Latihan 5

```
#!/bin/bash
clear
echo “nama login anda $LOGNAME”
echo “saat ini anda berada di direktori `pwd`”
```

```
echo “waktu sekarang adalah `date +%T`”
echo “selamat bekerja”
```

Latihan 6

```
#!/bin/bash
clear
echo “komputer anda telah menyala selama `uptime`”
echo “jumlah user yang login sebanyak `who | wc -l` user”
echo “anda login dengan user $LOGNAME”
echo “di shell $SHELL”
```

Latihan 7

```
#!/bin/bash
clear
data='date +%D'
jumlah=`who | wc -l`
echo “tanggal $data ada $jumlah user yang login”
```

Latihan 8

```
#!/bin/bash
user=andi
echo “hai $user i'm glad to meet you”
echo hai $user i\#m glad to meet you
echo “apakah kamu punya teman “special”?”
echo “\”special\” ?? teman apa itu?”
echo “ya pokoknya “special””
```

Latihan 9

```
$ function cetak_selamat {
> echo “Selamat Datang”
> echo “Di Shell Programming”
> }
$ cetak_selamat
Selamat Datang
Di Shell Programming
```

Latihan 10

```
$ function cetak_selamat () {
> echo “Selamat Datang”
> echo “Di Shell Programming”
> }
$ cetak_selamat
Selamat Datang
Di Shell Programming
```

Latihan 11

```
$ data=”halo ini bejo”
$ sdata=${data#*lo}
$ echo $sdata
$ tdata=${data%be*}
$ echo $tdata
```

Latihan 12

```
#!/bin/bash
TITEL=”Membuat Fungsi Sistem Informasi $HOSTNAME”
SAAT_INI=$(date+”%d %T %Z”)
UPD=”Sistem ini di update oleh $USER pada tanggal
$SAAT_INI”
function info_uptime(){
   echo “<h2>informasi uptime</h2>”
   echo “<pre>”
   uptime
   echo “</pre>”
}
cat <<- EOF
<HTML>
<HEAD>
<TITLE>$TITEL</TITLE>
</HEAD>
<BODY>
<H1>$TITEL</H1>
<P>$UPD</P>
$(info_uptime)
</BODY>
</HTML>
EOF
```

Latihan 13

```
#!/bin/bash
let data1 data2 hasil
read --p “masukkan sebuah angka : “
data1=$REPLY
read --p “masukkan sebuah angka lagi : “
data2=$REPLY
((hasil=data1-data2))
echo “hasil dari $data1 - $data2 adalah $hasil “
unset data1 data2 hasil
```

3. analisa kode berikut, apakah sudah benar jika belum benarkan

```
while :
 do
   clear
   echo "--------------------------------------"
   echo " Main Menu "
   echo "--------------------------------------"
   echo "[1] Show Todays date/time"
   echo "[2] Show files in current directory"
   echo "[3] Show calendar"
   echo "[4] Start editor to write letters"
   echo "[5] Exit/Stop"
   echo "======================"
   echo -n "Enter your menu choice [1-5]: "
   read yourch
   case $yourch in
```

```
        1) echo "Today is date , press a key. . ." ; read ;;
        2) echo "Files in pwd" ; la; echo "Press a key. . ." ; read ;;
        3) cal ; echo "Press a key. . ." ; read ;;
        4) vi ;;
        5) exit 0 ;;
        *) echo "Opps!!! Please select choice 1,2,3,4, or 5";
           echo "Press a key. . ." ; read ;;
    esac
done
```

IV. Tugas

1. Buat skrip untuk melihat aktifitas jaringan pada sistem anda.
2. Buat skrip untuk merubah input lowercase menjadi uppercase.
3. Cetak variable i = 1 to 20, dengan ketentuan 1..10 jalan pada baground sehingga outputnya :
   11 12 13 14 15 16 17 18 19 20
   1 2 3 4 5 6 7 8 9 10
4. Tuliskan sebuah program dan terdapat kontrol dimana yang bisa mengeksekusi program tersebut hanyalah 'root'.
5. Buat program yang mematikan dirinya sendiri.
6. Buat program yang menghapus file disebuah direktory jika file itu mengandung bad karakter (/[\ +\{\;\"\\\=\?~\(\)\<\>\&*\|\$]/p`)

selamat mengerjakan