

LAB 10

Course Code: CSC 2209

Course Title: Operating Systems



Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	10	Week No:	10	Semester:	
Lecturer:	<i>Name & email</i>				

Lecture Outline



1. Loop
2. While Loop
3. For Loop
4. For Loop With Commands
5. Select Loop With Cases
6. Until Loop
7. Break and Continue

Loop

- ☐ There are 4 type of loops in shell scripting.
 - ☐ The While Loop
 - ☐ The for Loop
 - ☐ The select loop
 - ☐ The until loop

While Loop

☐ Syntax

```
while [ condition ]  
do  
    statement  
    statement  
done
```

☐ Example

```
n=1  
while [ $n -le 10 ]; do  
    echo $n  
    n=$(( n+1 ))           # n=$(( n+1 )) or (( n++ )) or (( ++n ))  
done
```

For Loop

❑ Syntax

```
for VARIABLE in 1 2 3 4 5 .. N [We can use file also]
```

```
do
```

```
    statement
```

```
done
```

```
for OUTPUT in $(unix command)
```

```
do
```

```
    statement
```

```
done
```

```
for (( initiate; condition; increment/decrement ))
```

```
do
```

```
    statement
```

```
done
```

Example

```
for number in 1 2 3 4 5; do
    echo $number
done
```

*****The following syntax will work on bash version 4 or above*****

```
for number in {1..100} ; do          # [This will loop through 1 to 100]
    echo $number
done
```

```
for number in {1..100..2}; do        #[This will increment by 2]
[START..END..INCREMENT]
    echo $i
done
```

Example

❑ The familiar one

```
for (( i=0; i<10; i++ )); do
```

```
    echo $i
```

```
done
```

For Loop With Commands

```
for command in ls pwd date; do
    echo "Command Name: $command" #[with echo it will not execute]
    echo "Command Output:"
    $command                      #[for executing only write the command]
done
```


Select Loop With Cases

```
select variable in alal dulal rahim
do
    case $variable in [switch]
        alal ) [case alal]
            echo Alal is selected
        ;;
        dulal ) [case dulal]
            echo Dulal is selected
        ;;
        rahim ) [case rahim]
            echo Rahim is selected
        ;;
        * ) [default]
            echo Default
        ;;
    esac
done
```

- ☐ • This will take inputs infinitely
- ☐ • Press Ctrl+C to exit

Until Loop

☐ Syntax

```
until [ condition ] ; do  
    statement  
done
```

☐ Example

```
#!/bin/sh  
a=0  
until [ ! $a -lt 10 ]; do  
    echo $a  
    a=`expr $a + 1`  
done
```

Break and Continue

☐ Break

```
for (( i=1; i<10; i++ )); do
    if [ $i -gt 5 ]; then
        break
    fi
    echo "$i"
done
```

☐ Continue

```
for (( i=1; i<10; i++ )); do
    if [ $i -eq 5 ]; then
        continue
    fi
    echo "$i"
done
```



Books

- ❑ Unix Shell Programming
 - ❑ Written by Yashavant P. Kanetkar