AOSS
TRAINING

OSS Applications

Topics

1. Using LAMP
2. MySQL
3. PHP
4. Web Application -Drupal

Using LAMP

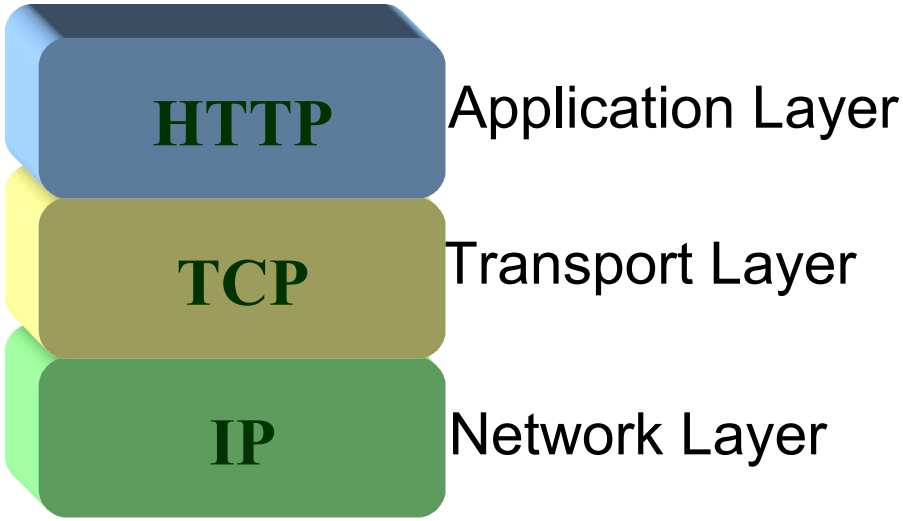
AOSS
TRAINING

Web Server Overview

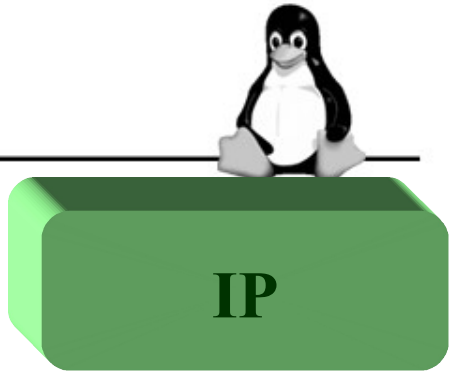
- To serve HTML pages
- HTTP
  - Communication mechanisms with the client
- Provides application framework
  - PHP
  - Common Gateway Interface (CGI)

Basic Networking Concepts

- TCP/IP protocol



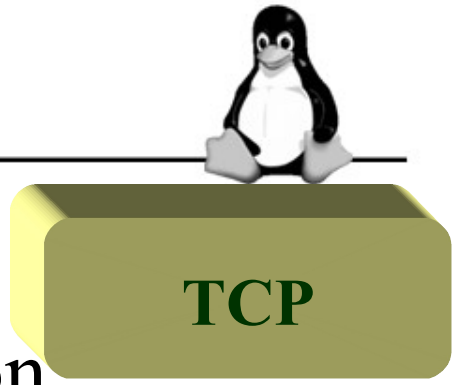
Internet Protocol



- *Unique Identifier* or a*ddress*
- To identify a computing device (e.g. your PC) on the Internet
- To deliver packet from one end to the other end
- Four bytes address separated by a '.'
  - byte1.byte2.byte3.byte4
  - Each byte's value range from 0 to 255
  - E.g. 192.168.1.100

AOSS TRAINING

Transmission Control Protocol (TCP)



- Provides reliable data transmission
  - Error checking
  - Retransmission on error
  - No missing data
- Relies on IP for addressing and packet delivery
- Application is identified by Port Number
  - E.g. 80 (Web server TCP port number)
  - E.g. 25 (Email server)

AOSS TRAINING

Hypertext Transfer Protocol (HTTP)

- Communication mechanisms for browser and web server

- Relies on TCP
  - HTTP data transfer is reliable
- Request and response
- Stateless
  - Current connection is not related to the previous connection

AOSS TRAINING

Domain Naming System (DNS)

- Domain name
  - Name under one administrative control
  - E.g. “codemovers.org”
- Host name
  - A name of a system within the domain (computer name)
  - E.g. “www”
- Fully Qualified Domain Name (FQDN)
  - Host name + Domain name
  - E.g. “www.codemovers.org”

AOSS TRAINING

Domain Name System (DNS)

- Remembering IP address is difficult
- Need to associate a name with an IP address
  - E.g. 192.168.1.100 >> test.codemovers.org
- DNS server
  - Maps a fully qualified domain name to its IP address

Web Server

- A program that
  - Accepts HTTP request
  - Sends HTTP response
  - Serves document (mostly HTML)

Web Server Extensions

- Most web server support extensions
  - Extends HTTP functionalities
  - Added features
- SSL and TLS
  - Provides secure transaction
- WebDAV
  - Read, write and version document on the web server

Apache HTTP Server

- A web server
- Developed by Apache Software Foundation
- Apache License
  - Open Source
  - Free (gratis and freedom)
  - Reasonable conditions

AOSS TRAINING

Working with Apache HTTP Server

- Installation
- Configuration
- Testing
- Deployment

Installation on Linux

- Packages
  - RPM
  - DEB
- Compile
  - Manual compilation
  - Most flexible

Configuration

- Configuration file
  - Directives
  - Text file
  - Read once during startup
- Example
  - /etc/httpd/conf/httpd.conf

Directives

- Control Apache HTTP server operations
- Key important directives
  - ServerName
  - Listen
  - ServerRoot
  - DocumentRoot
  - User
  - Group

ServerName

- For testing
  - Use: localhost
- For real deployment
  - Fully Qualified Domain Name
  - E.g. www.apache.org

Testing

- Check if the configuration file is correct
  - apachectl –t
- Run the server
- Create test file (index.html)
- Connect to TCP port 80
  - Browser
  - Telnet

Web Application

- Common Gateway Interface (CGI)
  - Executes program on the web server
  - Standard techniques on passing data from client to the server
  - Programming language neutral
- Direct Module Interface
  - PHP
  - Mod_perl
  - Apache Tomcat

PHP as Apache Module

- Setting PHP as Apache's module
  - In the Apache's *configuration* file, add:

```
LoadModule php5_module libexec/libphp5.so
AddType application/x-httpd-php .php
```

- Line 1: load the PHP5 module
- Line 2: Any document request with *.php* extension would be handled by the PHP module

Exercise 1

- Installation

- /* In Debian */
  # apt-get install apache
- /* In Fedora Core 4 (only Apache 2.x is supported) */
  # yum install httpd

Exercise 2

- Testing

- # /etc/init.d/apache start

Exercise 3

- Configure DocumentRoot (edit /etc/apache/httpd.conf)

- /**
  Use any editor to open ***/etc/apache/httpd.conf***
  Look for the following ***DocumentRoot*** *directive* and change it to the following
  **/
  DocumentRoot /home/knoppix/www
- /**
  Restart the server.
  **/
  # **/etc/init.d/apache restart**

Exercise 3

- Create index.html

```
<html>
<body>
<h1>Hello from new DocumentRoot</h1>
</body>
</html>
```

Exercise 4

- Apache Logs

- To see the content of the log file, say **access.log**.
- # cat /var/log/apache/access.log

Exercise 5

- Configure CGI (edit /etc/apache/httpd.conf)

```
ScriptAlias /mycgi/ /home/knoppix/cgi/
   <Directory /home/knoppix/cgi>
     AllowOverride None
     Options ExecCGI
     Order allow,deny
     Allow from all
   </Directory>
```

Exercise 5

- Create test.sh

```
#!/bin/bash
echo "Content-type: text/plain“
echo

env
```

Exercise 6

- Configure PHP
- **# cd /etc/apache**
  # cat modules.conf
  ...
  LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
- # **cd /etc/apache/conf.d**
  # **cat php4.conf**
  <IfModule mod_php4.c>
  AddType application/x-httpd-php .php .phtml .php3
  AddType application/x-httpd-php-source .phps
  </IfModule>

Exercise 6

- Create test.php

```
# cd /home/knoppix/www
# echo "<? phpinfo(); ?>" > test.php
```

Exercise 7

- Configure MySQL

- # /etc/init.d/mysql start

Lab 8

- MySQL and PHP
- **# grep mysql /etc/php4/apache/php.ini**

  extension=mysql.so

MySQL

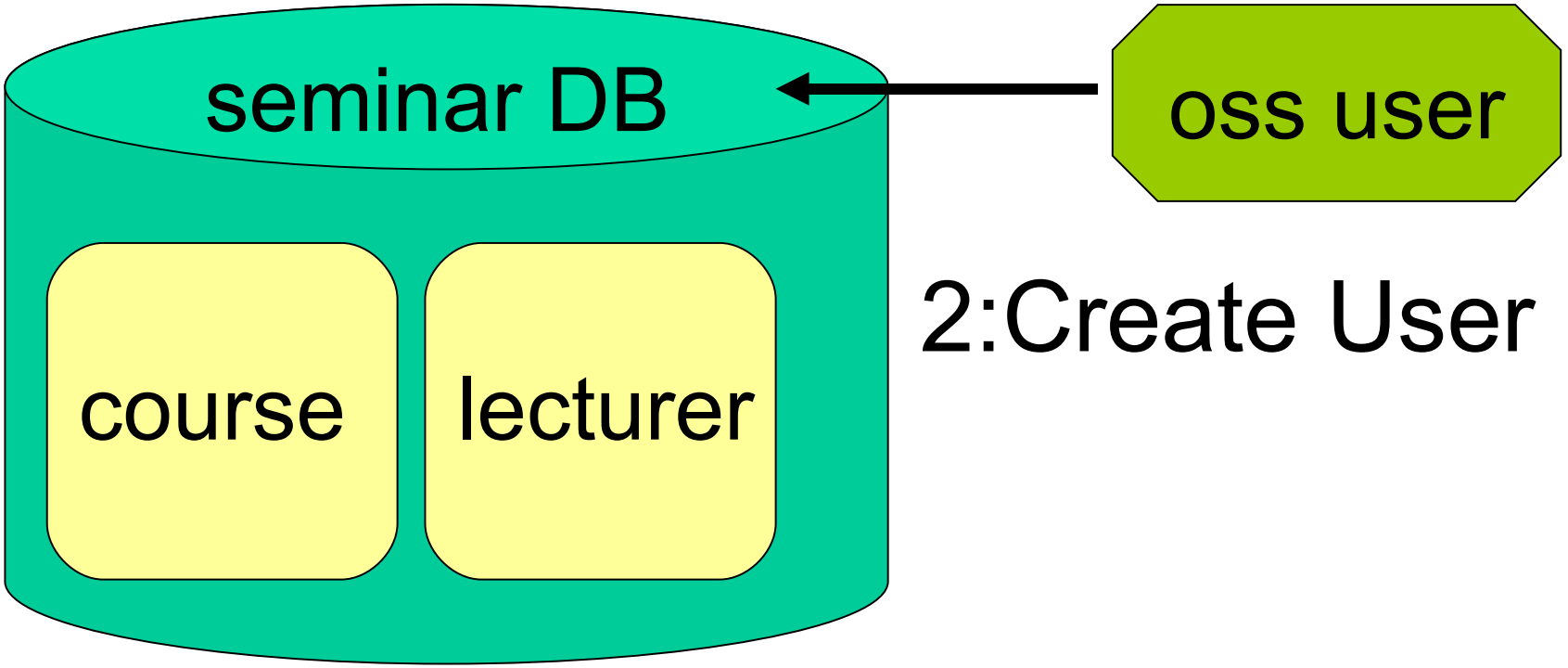
AOSS
TRAINING

At the end of this session, you will be able to :

- Create MySQL Database
  - Create user, table and load data
- Describe special feature of MySQL
  - Create InnoDB table

Create Database Step by Step

1:Create Database



2:Create User

3:Create Table and load data

Exercise 1

- Start up Terminal
- Create “seminar” database;

Exercise 1

- $ sudo -s
- $ /etc/init.d/mysql start
- $ mysqladmin create seminar

Change root password

```
SET PASSWORD FOR
root@localhost=
        PASSWORD('new_password');
```

Reload the grant tables

Flush privileges

Create Database

- CREATE DATABASE *db_name*

- Example

  Create database seminar

Create User

- By using GRANT statements

  GRANT *privileges_type*
  ON *db_name. table_name* TO *user*
  IDENTIFIED BY '*password*'

- Example

  GRANT ALL ON seminar.* TO oss@localhost
  IDENTIFIED BY 'secrets';

Table Type

- TST

Transaction-safe tables

InnoDB

BDB

- NTST

Not transaction-safe tables

MyISAM

ISAM

HEAP(MEMORY)

MERGE

Create Table statement

- Create a new table within a database

CREATE TABLE *table_name*

(*column_name type*

[NOT NULL | NULL]
[DEFAULT *default_value*]
[AUTO_INCREMENT] [PRIMARY KEY]

) TYPE= *type_name* ;

AOSS TRAINING

Column definition

- Column type
  - INT[(*M*)]
  - CHAR(M)　　VARCHAR(M)
  - TEXT[(*M*)]
  - DATE　TIME　DATETIME
- Option
  - AUTO_INCREMENT
  - NOT NULL
  - DEFAULT
  - PRIMARY KEY

AOSS TRAINING

Create “lecturer” table in “seminar” database

- Create “lecturer” table

Column name	Type	Size	Option
id	int	2	PRIMARY KEY
name	varchar	20	

- Example

```
Use seminar;
CREATE TABLE lecturer (
 id INT(2),   name VARCHAR(20),
 PRIMARY KEY (id)  );
```

AOSS TRAINING

Exercise 2
Create “course” table

- Create “course” table

Column name	Type	Size	Option
course_id	int	2	AUTO_INCREMENT PRIMARY KEY
title	varchar	20	NOT NULL
lecturer_id	int	2	

AOSS TRAINING

Exercise 2
Create "course" table

- CREATE TABLE course (
  course_id int(2) AUTO_INCREMENT,
  title varchar(20) NOT NULL,
  lecturer_id int(2),
  PRIMARY KEY(course_id)
) ;

AOSS TRAINING

Load data to “lecturer” table INSERT Statement

Lecturer_id	Name
1	Jack
2	Jill

- INSERT INTO table_name [(column1, column2...)]

  VALUES(value1, value2...)

- Example
  INSERT INTO lecturer VALUES (1,’Jack’)

AOSS TRAINING

Load data to “course” table

course_id	title	lecturer_id
	Web Server	2
	MySQL and PHP	2
	OpenOffice	1

- Example
  INSERT INTO course VALUES (Null,’Web Server’,2)

AOSS TRAINING

Retrieve rows from table SELECT Statement

- SELECT *column1* [,*column2*]...

  FROM *table_name* [,*table_name2*]...

  [WHERE *condition*] [ORDER BY *column1* [,*column2*]...]

- Example

SELECT * FROM course

SELECT * FROM course WHERE lecturer_id = 2;

SELECT * FROM lecturer WHERE Name = ‘Jill’;

SELECT * FROM lecturer WHERE Name Like ‘J%’;

AOSS TRAINING

Table Type

InnoDB

- TST (Transaction-safe tables)
- Support Foreign Key
- Support Rollback data

MyISAM

- NTST (Not transaction-safe tables)
- Not support Foreign Key
- Not support Rollback

AOSS TRAINING

FOREIGN KEY

FOREIGN KEY [id] (index_column_name, ...)
REFERENCES table_name (index_column_name ...)

"Lecturer" table

id	Name
1	Jack
2	Jill

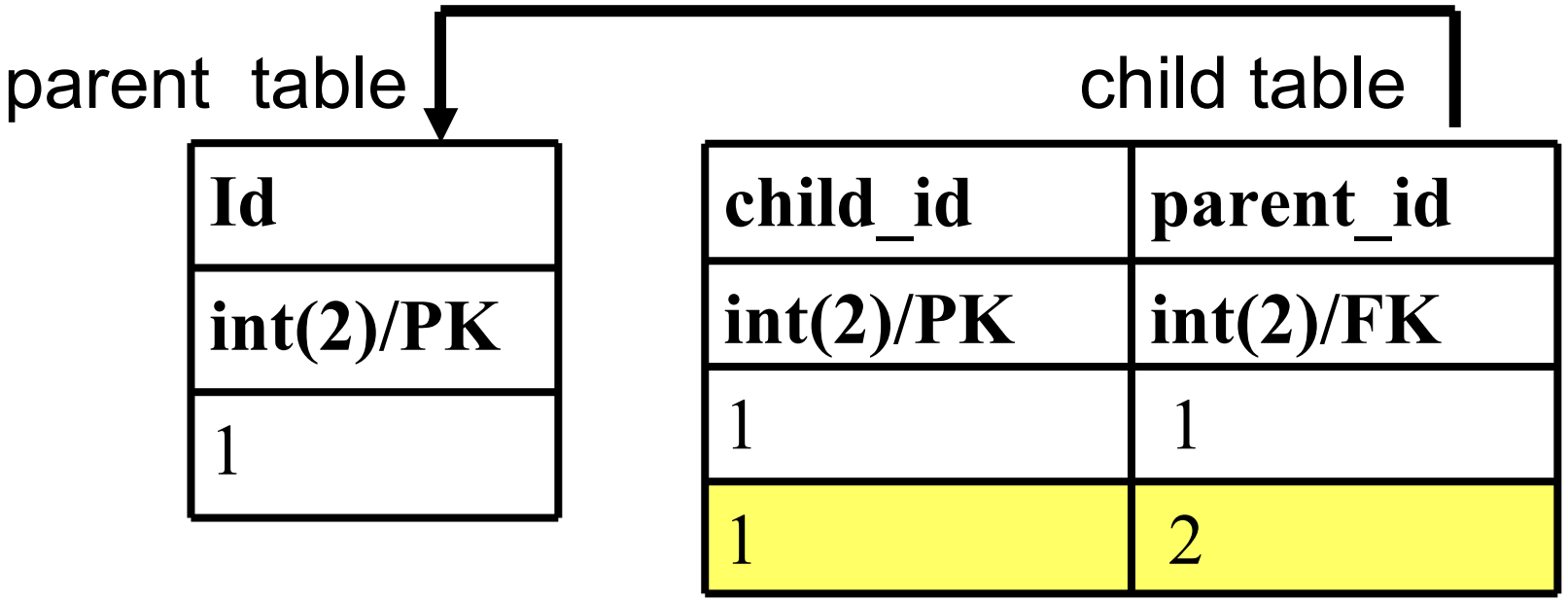
"Course" table

Course_id	title	Lecturer_id
1	--	2
2	--	2
3	--	1

CREATE TABLE course (course_id int(2), -------,
PRIMARY KEY(course_id),
FOREIGN KEY (lecturer_id) REFERENCES leturer(id))
Type=InnoDB

AOSS TRAINING

Exercise 3
(Test for InnoDB)



** parent table and child table must be same type

AOSS TRAINING

Exercise 3 (Test for InnoDB)

```
CREATE TABLE parent (id INT NOT NULL,
   PRIMARY KEY(id) ) TYPE=INNODB;

CREATE TABLE child
 (child_id INT, parent_id INT,
FOREIGN KEY (parent_id)
      REFERENCES parent(id) )
TYPE=INNODB;
```

AOSS TRAINING

Data file

- InnoDB

Mysql folder
  |-- ibdata1
  |-- ib_arch_log_0000000
  |-- ib_logfile0
  |-- ib_logfile1
  |--database_name folder
        |-- child.frm
        |-- parent.frm

- MyISAM

Mysql folder
  |--database_name folder
        |-- child.frm
        |-- child.MYI
        |-- child.MYD
        |-- parent.frm
        |-- parent.MYI
        |-- parent.MYD

AOSS TRAINING

Exercise of Transaction with InnoDB

- select * from parent;
  // show the original data;
- begin;
  // begin transaction;
- insert into parent values(10);
  // add new data
- select * from parent;
  // you can see the data is added
- Rollback;
  // rollback to beginning of transaction
- select * from parent;
  // see what happen ??

AOSS TRAINING

MySQL Exercise 4-1

- Create new database “customer”
- Create new user “admin” for the “customer” database

Country_info

id	name
60	Singapore
65	Malaysia

Customer_info

customer_id	country_code
1	60
2	65

AOSS TRAINING

MySQL Exercise 4-2

- Create table “country_info” and insert some data.

Column name	Type	Size	Option
id	int	2	PK
name	varchar	20	NOT NULL

AOSS TRAINING

MySQL Exercise 4-3

- Create table “customer_info” and insert some data.

Column name	Type	Size	Option
id	int	2	PK
name	varchar	20	NOT NULL
country_code	int	2	NOT NULL / FK
DOB	date		
TEL	varchar	20	
contact_type	int	1	Default 1

AOSS TRAINING

PHP

At the end of this session, you will be able to :

- Basic PHP Programming
  - –Variable, Operator, Control structures
  - –Form processing
- Access to MySQL Database from PHP
  - –Data access programming

PHP Pages

- HTML with PHP code embedded in it
- PHP code is enclosed in

```
<?php
        // PHP code here
?>
```

- Short form:

```
<?
        // PHP code here
?>
```

Variable

- Do not need declaration
- Starts with a “$” sign
- Example
  - `$a = 1000;`
  - `$b = 1.234;`
  - `$c = “Jack”;`
  - `$d = ‘Jill’;`

Single vs. Double quotes

- Double quotes **evaluates** the variable within the quotes
- Single quotes does not
- Example:

```
$a = “it”;
$b = “That is $a”;  // That is it
$c = ‘That is $a’;  // That is $a
```

Single vs. Double quotes

```
<?php

$a = “it”;
$b = “That is $a”;
print “$b”;
$c = ‘That is $a’;
print “$c”;

?>
```

Output as HTML

- Use the `print` or `echo` statement
- Example:

```
<HTML><BODY>
<?php
    print("Hello world!");
     print "Hello world!<br>";
     echo "<h1>Hello world!</h1>";
?>
</BODY></HTML>
```

AOSS TRAINING

Operators

+ - / * %	Arithmetic
.	String
++ --	Increment, decrement
== !=	Equality and inequality
!	Logical NOT
&& ||	Logical AND and OR
< > <= >=	Less or greater than
+= -= *= \=	Shortcuts

AOSS TRAINING

Operator

- String Operator
- . operator
- Combine (concatenate) two strings into one
- Example:

```
$a = “Tux “;
$b = “The Penguin”;
$c = $a . $b;
// $c = “Tux the Penguin”
```

- Arithmetic Operators
- Example

```
$a = 0;
$a = $a + 1;
   // $a = 1
$a++;
   // $a = $a + 1
$a--;
   // $a = $a – 1
$b = 5 % 2;
   // $b = 1
```

AOSS TRAINING

Branching & Loop

- Branching

```
if (condition 1)
    // statement one
elseif (condition 2)
    // statement two
else
    // statement three
```

- Loop

```
for (initial; condition; counter)
{
    // statements
}
```

```
while (condition)
{
    // statements
}
```

Branching & Loop

- Branching

```
if (condition 1):
   // statement one
elseif (condition 2):
   // statement two
else
   // statement three
endif;
```

- Loop

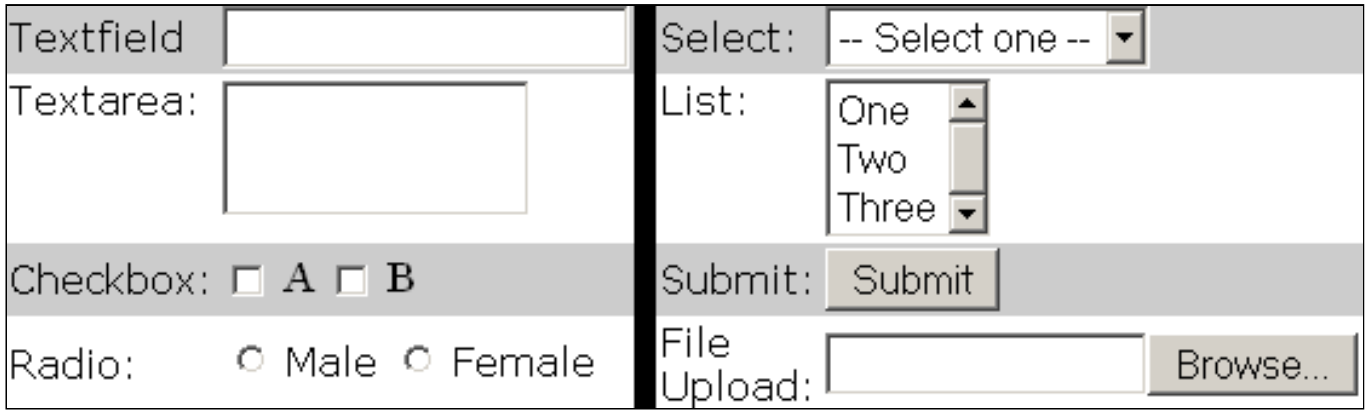
```
for (initial; condition; counter):
    // statements
endfor;
----
while (condition):
   // statements
endwhile;
```

PHP Form Processing

- Targeted server-side PHP program should handle the data
- The target PHP program is specified by `ACTION` attribute
- Three predefined array to access form data:
  - $_GET – submitted by GET method
  - $_POST – submitted by POST method

Form Objects

- Form objects on browser



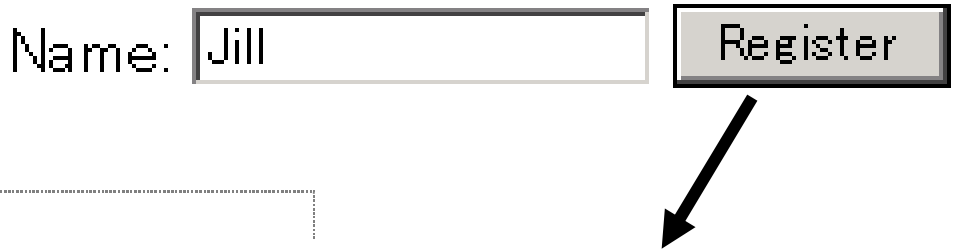
AOSS TRAINING

PHP Form Processing

- Exa

form.html

```
<form method="POST"  action="register.php">
Name: <input type="text" name="NameText">
<input type="submit" value="Register">
</form>
```



register.php

```
<?php
  $name = $_POST['NameText'];
  print("Hello $name");
?>
```

Hello Jill

form.html

```
<HTML><HEAD>
<TITLE> PHP FORM PROCESSING </TITLE><HEAD>
<BODY>
<FORM METHOD="POST"  ACTION="register.php">
Name: <INPUT TYPE="text" NAME="NameText">
<INPUT TYPE="submit" VALUE="Register">
</FORM>
</BODY>
</HTML>
```

register.php

```
<?php
  $name = $_POST['NameText'];
  print("Hello $name");
?>
```

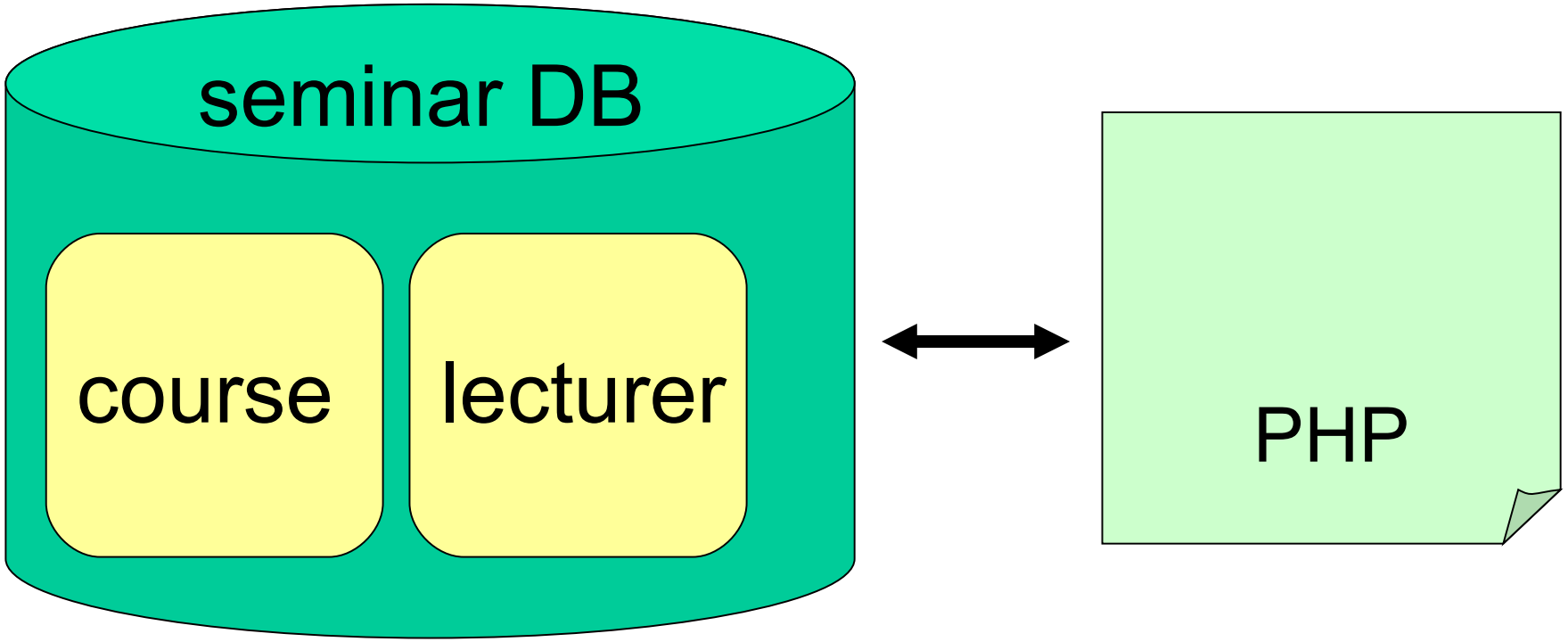
AOSS TRAINING

MySQL and PHP

- Access to MySQL Database from PHP
  - Data access programming

Accessing MySQL Database from PHP Program

AOSS TRAINING



1: Access to “lecturer” table and show data list.

AOSS TRAINING

MySQL and PHP

- Using PHP you can connect and use MySQL
- Steps:

  (1) Open database connection

  (2) Select your database

  (3) Run the query (using SQL)

  (4) Retrieve the results

  (5) Close the database connection (optional)

Create DB Access Program (1)

(1) mysql_connect (*host, name, password*)

(2) mysql_select_db (*dbname [, conn]*)

(3) mysql_query(*sql_string [,conn]*)

Create DB Access Program(1)

(4) mysql_fetch_array (*result [,restype]*)

- echo "user_id: ".$row[“lecturer_id"]."
";
- echo "user_id: ".$row[0]."
";

- echo "fullname: ".$row["Name"]."
";
- echo "fullname: ".$row[1]."
";

(5) mysql_close(*[conn]*)

dbsample1.php

```
 1: <?php
 2: $host = "localhost";
 3: $user = “oss";
 4: $pass = “secret";
 5: $database=“seminar”;
 6:
 7: $conn = mysql_connect ($host, $user, $pass)
 8:
 9: mysql_select_db($database,$conn );
10:
11: $sql_query = "SELECT * FROM lecturer ";
12: $result = mysql_query($sql_query);
13:
14: while ($row = mysql_fetch_array ($result)) {
15:  echo "user_id: $row[0] <br>";
16:  echo "fullname: $row[1] <br>";
17: }
18: mysql_close();
19: ?>
```

Create DB Access Program(2)

- Steps:
  (1) Open database connection
  (2) Select your database
  (3) Run the query (using SQL)
  (4) If error happened

      Show error

    Else

      Show the result using table

  (5) Close the database connection (optional)

AOSS TRAINING

dbsample2.php

```
1: <?php
2: $host = "localhost";
3: $user = “oss";
4: $pass = “secret";
5: $database=“seminar”;
6:
7: $conn = mysql_connect ($host, $user, $pass) or die ('I cannot connect to the database because: ' .
     mysql_error());
8:
9: mysql_select_db($database,$conn );
10:
11: $sql_query = "SELECT * FROM lecturer ";
12:
13: $result = mysql_query($sql_query);
14: if ($result == 0)
15: {
16:     echo "I had a problem running the query!";
17: }
18: else
19: {
```

AOSS TRAINING

dbsample2.php

```
20:    print("<TABLE border=1>");
21:    while ($row = mysql_fetch_array ($result)) {
22:       print("<TR>");
23:       for ($j=0; $j<2; $j++) {
24:          print “<TD> $row[$j] </TD>";
25:    }
26:       print("</TR>");
27:    print ("</TABLE>");
28: }
29: mysql_close();
30: ?>
```

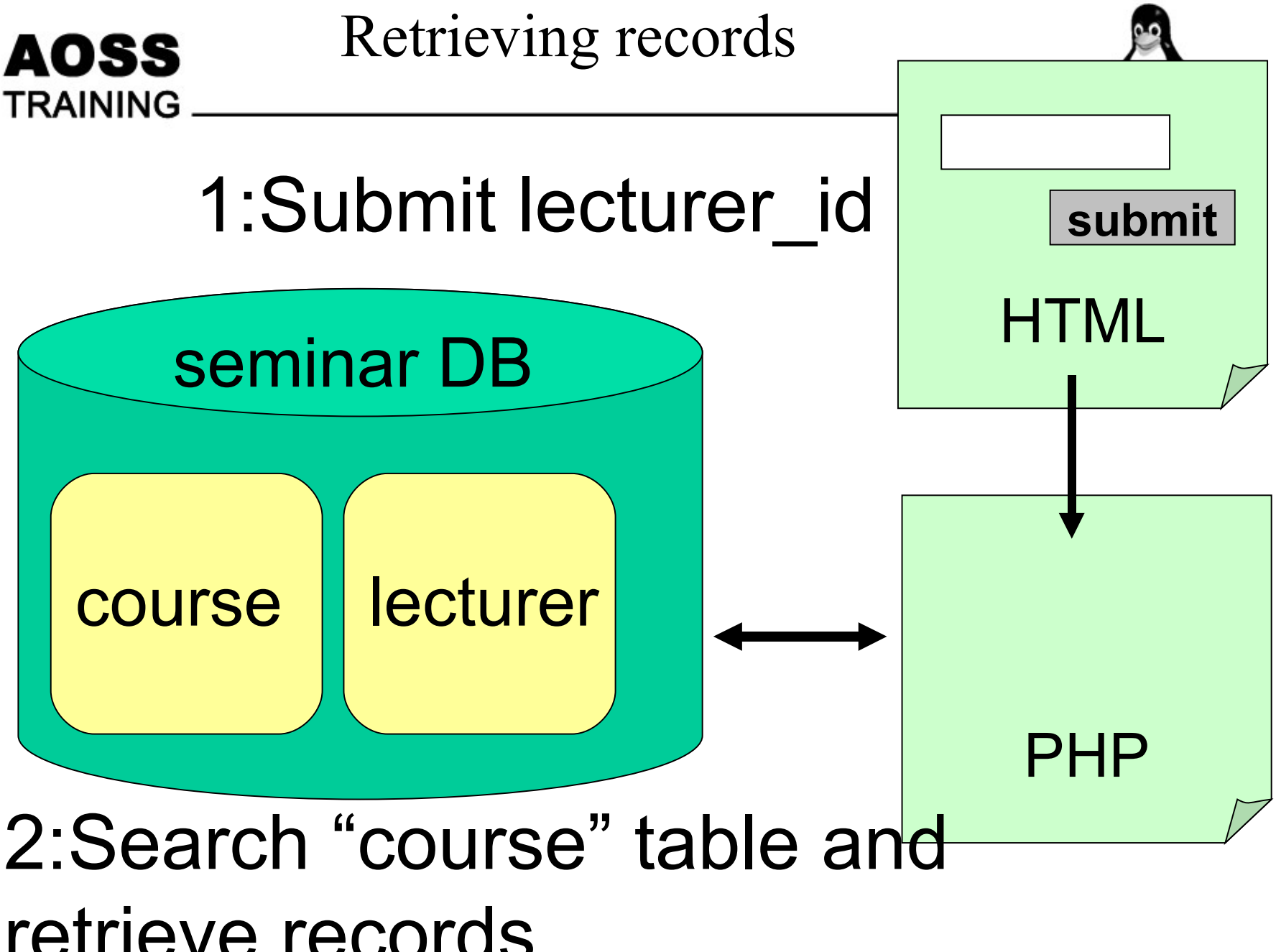
Retrieves the number of rows

```
$num_rows = mysql_num_rows($result);
print "$num_rows Rows";
```

This command is only valid for SELECT statements.

AOSS TRAINING

Retrieving records



AOSS TRAINING

Submit data to PHP

- HTML (Mytest1-1.html)

<FORM action="Mytest1-2.php" method="**post**">
Lecturer ID <INPUT TYPE="text" NAME="**lid**" SIZE="8">

- PHP (Mytest1-2.php)

$lid = $_**POST**['**lid**'];

$sql_query = "SELECT ---- WHERE lecturer_id =" . $lid ;

AOSS TRAINING

Simple Input Validation

- isset(var) Determine whether a variable is set

- is_string(var) Finds whether the given variable is a string
- is_numeric(var) Finds whether a variable is a number or a numeric string

AOSS TRAINING

Exercise
Insert New Data

- Example

Insertform.html

```
<form method="POST"  action="Insert.php">
Lecturer ID: <input type="text" name="lid"><BR>
Lecturer Name: <input type="text" name="lname"><BR>
<input type="submit" value="Register">
</form>
```

Insert.php

```
$lid=$_POST['lid'];
$lname=$_POST['lname'];
$sql_query = "INSERT INTO lecturer VALUES ($lid, '$lname')");
$result = mysql_query($sql_query);
```

AOSS TRAINING

Exercise MySQL & PHP 1

- Create trainee table and insert new data as following:

TraineeID	**TraineeName**	**Country**	**ContuctNumber**
Integer Size:3 Primary Key	Varchar Size:50 NotNull	Varchar Size:20 NotNull	Varchar Size:20
1	Mohamad	Malaysia	60-12345678
2	Yati	Indonesia	62-23456789

- Create HTML/PHP program to insert new data and retrieve all data from trainee table

Exercise MySQL & PHP 2

- Create simple program to insert data to course table.

- As MyISAM table type is used (not concern Foreign Key), You may need to add extra code to check whether lecturer id has already existed or not.

Web Application -Drupal

AOSS
TRAINING

AOSS TRAINING

Agenda

- Overview
- Drupal
- Getting Drupal
- Installing Drupal
- Adding Modules

Overview

- Web Server recap
- MySQL & PHP recap
- Installation of a typical PHP based web based application

Drupal

- Drupal is a Content Management System
  - Current version 4.6.2
  - http://www.drupal.org
  - Features
    - www.CMSMatrix.org

Getting Drupal

- Download a copy of drupal from
- http://www.drupal.org
  - Click on Download
  - Click on 4.6
  - Obtain the latest version
    - Current version as of writing is 4.6.2

Installing Drupal

- Step 01 -Identify the DocumentRoot
  - Locate the apache2.conf or httpd.conf file of the apache server
  - Locate the DocumentRoot from the file

Installing Drupal

- Step 02 . Unzipping the drupal tarball
  - Unzip the drupal tarball
    - Run the command
      - tar xfj drupal-4.6.2.tar.bz2
  - Click on Kmenu -> Utilities -> Ark
    - Click on the file

Installing Drupal

- Step 03 . Reading the INSTALL.txt file
  - Using the text editor to read INSTALL.txt
    - Click on Kmenu -> Editors -> Advanced Text Editor (Kate)
    - Select the INSTALL.txt file

Installing Drupal

- Step 05 . Loading the Database Scheme
  - Run the command
    - mysql -uroot -p{password} drupaldb < database/database.mysql

Installing Drupal

- Step 06 . Connecting Drupal to the MySQL
  - Edit the file sites/default/settings.php
  - $db_url = "mysql://username:password@localhost/database" ;
  - $base_url =”http://localhost/";

Installing Drupal

- Step 07 . Configure Drupal
  - Create a “files” directory
  - Ensure that this directory is readable and writeable by the user.

Installing Drupal

- Step 08 . Adding a cron job
  - Run the cron server
    - Cron is a scheduler that runs tasks at specific time.
  - Run the command “crontab -e”
  - Type in the command:
    - 0 * * * * wget -O --q http://localhost/cron.php

Installing Drupal

- Step 09 . Testing the installation
  - You will have to test the installation by running your web browser, and typing the following in the location bar

    http://localhost/
  - Pls raise your hands if you have an error.

AOSS TRAINING

Installing Drupal Extras

- You can download a whole set of addons
  - Modules
  - Themes
  - Language

Installing Drupal Extras

- Step 01 . Downloading a module
  - Get a module from http://www.drupal.org
  - Click on “Download->Modules”
  - Select the module to download, and click on the download link to download the file

Installing Drupal Extras

- Step 02 . Unzipping the module
  - Unzip the module in the modules directory of drupal with the command:
    - tar xfz <module>-4.6.tar.gz

Installing Drupal Extras

- Step 03 . Additional reading
  - You may want to read the additional instructions needed to run the module
  - You may need to install additional tables in the database. Use the command:
    - mysql -uroot -p{password} drupaldb < {sql filename}

Conclusion

- Students should be able to confidently read instructions and install a simple open source based php application