Technikerschule - Fachschule für
Maschinenbau-, Metallbau-, Informatik- und Elektrotechnik
der Landeshauptstadt München

Technikerprüfung 2013

Datenverarbeitungstechnik

Zeit : 150 Minuten

Klasse :	Name :

	Punkte:	Note :	Unterschriften:
Erstkorrektur			
Zweitkorrektur			

Teil 1 , ohne Unterlagen

Name, Klasse :

1. Erklären Sie den Unterschied zwischen präemptivem und kooperativem Multitasking :

   1.1 Wie funktionieren die Verfahren ?

   Kooperativ bedeutet, daß die Tasks selber den Scheduler steuern.
   Immer wenn eine Task nach „blocked“ geht, wird eine andere gestartet.
   Die alte geht, wenn I/O fertig, nach sleeping.

   Präemptiv bedeutet, daß ein Timer („Quantum“) den Scheduler steuert.
   Die Tasks haben keinen Einfluß.

   4 PUNKTE

   1.2 Welche Probleme können jeweils auftreten ?

   Beim kooperativen kann eine task, die nie auf blocked geht (z.b. weil sie abgestürzt oder in eineer Endlosschleife ist), das ganze System blockieren.

   Beim paräemtiven können durch ungünstige Umschaltzeitpunkte Fehler passieren: (-> Semaphore auf critical sections)

   4 PUNKTE

2. Im Multitasking ist die Speicherbehandlung ein wesentlicher Aspekt :

2.1 Was bedeutet “virtual memory“ ?

Der unzureichende Hauptspeicher (RAM) wird unterstützt, indem man Seitenweise (z.b. 4k) Speicherbereiche auf die Festplatte auslagert.

2 PUNKTE

2.2 Was ist ein “paging-error“ ?

Eine adressierte Adresse liegt auf einer Seite, die gerade ausgelagert ist

2 PUNKTE

3. Kreuzen Sie die Aussagen an, die für Windows-Netze richtig sind :

- O Mit Active Directory können Festplattenverzeichnisse effektiv durchsucht werden
- O Ein Domänencontroller verwaltet Internet-Domänen
- **X** Ein Mitgliedsserver nimmt nicht an der Domänenverwaltung teil
- **X** Windows benötigt in Layer 3 zwingend das IP-Protokoll
- **X** Die Benutzer-Anmeldedaten sind im Verzeichnisdienst gespeichert
- O Mit Scripting.FileSystemObject können Benutzer-Anmeldedaten verändert werden
- **X** Mit dem Remote-Desktop von Windows wird das Freigaberechtesystem umgangen
- **X** Der Scheduler von Windows Server arbeitet prioritätsgesteuert
- **X** Windows benutzt virtual memory
- **X** Das Windows-Betriebssystem basiert auf protected-memory Prozessoren
- O Mit NTFS-Rechten wird der Zugriff auf Festplattenfreigaben nicht beeinflußt
- O Mit Priorität = “Echtzeit“ ist eine Windows-Task deterministisch
- O In Windows-Netzen leistet der Server die Rechenleistung für alle im Netz benutzten Applikationen

6.5 PUNKTE (1/2 pro Frage)

4. Welche Funktion hat der cgi-Mechanismus bei einem Webserver ?

Ausgabedaten von cgi-Skritpen (meist PHP, von einem HTM;L-Form aufgerufen) werden an den aufrufenden Socket (IP/Port) umgeleitet

2 PUNKTE

5. Kreuzen Sie die Aussagen an, die für Kommunikation mit Webservern richtig sind :

X Jede Webkommunikation wird mit einem http-Request begonnen

O Der Requester antwortet meist mit dem Schicken einer HTML-Seite

O Mit DocumentRoot kann man einstellen, wo Apache empfangene Daten hinschreibt

O Mit GET werden Daten empfangen, mit POST verschickt

X FORM ACTION gibt an, welches Skript im Server die Daten verabreiten soll

O Mit „Seitenquelltext anzeigen“ kann man den php-Code lesen

O Server-sided Skripte werden vor Ablauf vom Server zum Client übertragen

X Jeder Zugriff auf Apache wird in access_log gespeichert

O Aus access_log kann auch der Username des Anfragers ausgelesen werden

O Durch die vielen Router im Web ist die Quell-IP nicht nachvollziehbar

X Die Extension .php statt .html für cgi-Skripte verringert die Rechenleistung im Server

X Mit <?php beginnt Apache, Codezeilen an den PHP-Interpreter zu leiten

6 PUNKTE

6. Kreuzen Sie die Aussagen an, die für SQL-Datenbanken richtig sind :

X Der primary key ist immer auch ein Schlüsselkandidat

O Der Schlüsselkandidat ist immer auch ein Primary key

O Inkonsistenz bedeutet, daß Daten nicht mehr gespeichert werden können

O Attribute sind die Namen der Tables in einer Databse

O Ein SQL-Server kann nur eine Database, aber darin viele Tables verwalten

O Ein SQL-Server (z.b. mySQL) benötigt zur Funktion den Apache Server

O Durch Redundanz werden Daten in SQL sicherer

O Statische Redundanz von Einträgen erhöht die Datensicherheit weiter

4 PUNKTE

Teil 2, mit Unterlagen

Name, Klasse :

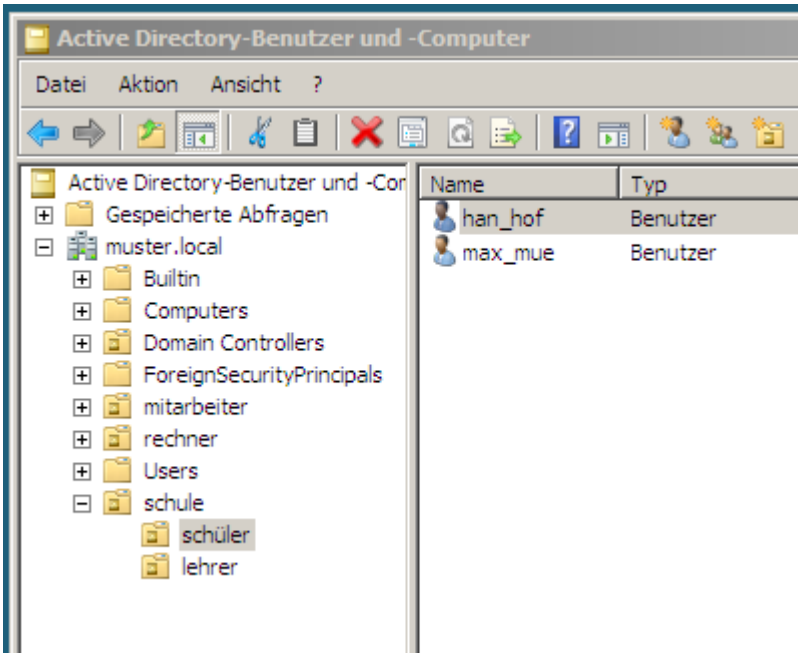
1. Aufgabe

Für Schülerkennungen in einem Schulnetz soll diese Funktion in VBscript programmiert werden :

In den Homedirectories aller Schüler wird, falls diese Datei noch nicht vorhanden ist, eine Datei **trash.dat** erzeugt.

Beispiel für ein Homedirectory : **//filer_labornetz/homes/max_mue**
Der Name des Homedirectories, und damit der Kennungsname, ist im Beispiel: **max_mue**

Hierzu werden die Schülerkennung nacheinander aus Active Directory ausgelesen :



Um auch in Active Directory den Kopiervorgang zu vermerken, wird dem Userobjekt, das als common name (cn) den Kennungsnamen hat (z.b. wie oben **max_mue**), nach dem Kopiervorgang das Attribut **description** mit dem Wert **„trash.dat ready“** beschrieben.
Dieses Attriut darf aber nicht zur Prüfung, ob die Datei existiert, verwendet werden !

1. Schreiben Sie ein VBscript, das die obige Funktion realisiert.

```
set dirobject = CreateObject("Scripting.FileSystemObject")
set adziel = GetObject("LDAP://ou=schüler,ou=schule,dc=muster,dc=local")
for each element in adziel
        a=element.name
        b=right(a,7)
        dirpfad = "c:/filer_labornetz/homes/"&b&"/trash.dat"
        if dirobject.FileExists(dirpfad) then
                msgbox("file schon vorhanden")
        else
                msgbox("file nicht vorhanden -> aktion")
                set fileziel =
                dirobject.CreateTextFile("c:/filer_labornetz/homes/"&b&"/trash.dat")
                usercontext = "LDAP://"&a&",ou=schüler,ou=schule,dc=muster,dc=local"
                set userziel = GetObject(usercontext)
                    userziel.put "description", "file_ready"
                userziel.setinfo
        end if
next
```

PUNKTE FÜR :
OU DC
for..each
wegen cn=
Pfad
FileExists
CreateTextFile
Pfad
a
put
setinfo

➔ **11 PUNKTE**

2. Aufgabe

Zur Archivierung Ihrer Sicherheitskopien von Mediendaten (Filme und Musik) auf Ihrer 2008-Servermaschine soll eine Weboberfläche mit unterlagerter PHP-Funktion geschrieben werden.

Ziel ist es, eine nach Typ geordnete Liste von Mediendateien aus ungeordnet gespeicherten Inhalten zu erzeugen.

Im Browser wird ein HTML-Form angezeigt, in dem Sie das Quellverzeichnis (z.b. d:\neue) und den gewünschten Datei-Typ (ohne Punkt) eingeben.

Als Datei-Tpen kommen : **mp3** und **wav** für Musik, sowie **avi** und **mpg** für Filme in Frage.

Das zu schreibende PHP-File soll nun alle Dateinamen (nur die Namen, nicht die Dateien !) vom jeweils gewünschten Datei-Typ von der Quelle lesen und je nach Typ entweder an die Datei **c:\media\my_films.txt** oder an **c:\media\my_music.txt** anhängen. (Beide Dateien exisitieren bereits.)

Nach Abschluß der Aktion soll die Anzahl der gefundenen Dateien vom gesuchten Typ am Browser angezeigt werden.

Zum Auslesen der Directory-Information verwenden Sie bitte den **readdir** – Befehl, dessen Bedienung im folgenden Beispiel gezeigt wird (hier werden die Dateinamen nur ausgegeben) :

Beispiel für readdir :

```
$handle=opendir("pfad");
while ($dateiname = readdir($handle))
        {
        echo "$dateiname";
        }
closedir($handle);
```

Das HTML-File sieht so aus :

```
<html><body>
Bitte geben Sie das Quelldirectory und den Datentyp an : <br><br>
<form action = “verwalt.php“ method = “get“>
      <input type =”text” name=”pfad” value=”Quelle”><br>
      <input type=”text” name=”typ” value=”Dateityp”><br>
      <input type=”submit” value=”Sortieren !”>
</form>
</body></html>
```

2. Schreiben Sie das php-Skript, das die obige Funktion realisiert.

```
<html><body>
<?php
        $pfad = $_GET[pfad];
        $typ = $_GET[typ];
        $handle = opendir($pfad);
        $zahl = 0;
        while($dateiname = readdir($handle))
                        {
                        $vorkommen=substr_count($dateiname,$typ);
                        if ($vorkommen > 0)
                            {
                            $zahl=$zahl + 1;
                            if ($typ == "mp3" OR $typ == "wav")
                                    {
                                    $liste=fopen("e:/my_music.txt","a");
                                    fwrite($liste,$dateiname."\r\n");
                                    fclose($liste);
                                    }
                            else
                                    {
                                    $liste=fopen("e:/my_films.txt","a");
                                    fwrite($liste,$dateiname."\r\n");
                                    fclose($liste);
                                    }
                            }

                        }
        print "gefundene dateien: ".$zahl;
?>
</body></html>
```

Punkte für :

$_GET

opendir

zähler

while..readdir

subst_count

incr

filetyp testen

append

liste schreiben

-> 9 PUNKTE

3. Aufgabe

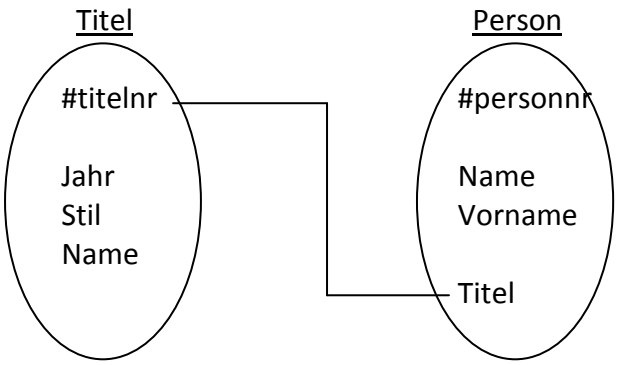
Die Funktion von Aufgabe 2 soll auf eine Lösung mt einer Datenbank umgestellt werden. Hierzu muß eine Database geplant werden, die Informationen über die gespeicherten Mediendateien enthält.

Gespeichert werden sollen Titel und Personendaten zu den jeweiligen Werken, wobei bei den Personen auch jeweils mehrere vorkommen können, bei einem Film z.b. Regisseur, Produzent, Hauptdarsteller usw.., bei einem Musikstück z.b. Komponist, Interpret usw.

Im Weiteren soll das Erscheinungsjahr und die Stilrichtung (z.b. Hip-Hop) gespeichert werden.

Es ist darauf zu achten, daß die Info auch vernünftig wieder gelesen werden kann, so sollte zum Beispiel erkennbar sein, welche Funktion eine Person bei einem bestimmten Werk innehatte.

3.1. Geben Sie in ER-Form 2 Tables hierfür an : **Titel** und **Personen,** zunächst vereinfacht, ohne die geforderten Informationen vollständig speichern zu können.



Punkte für : primary key

Attribute

Relation

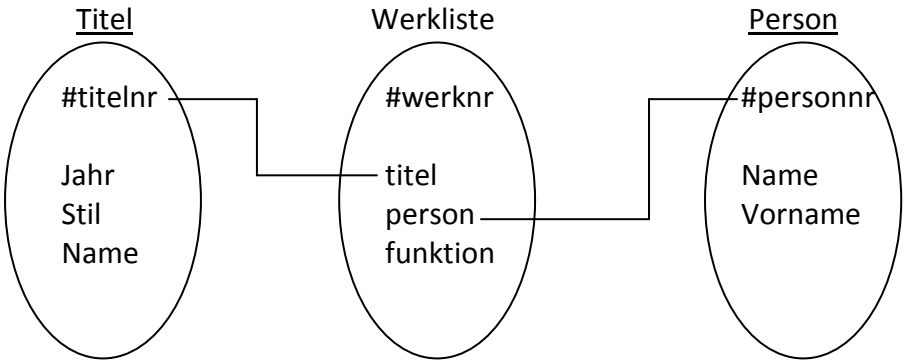
n:m !

n:m erkannt

-> 4 PUNKTE

3.2. Prüfen Sie mit dem ER-Modell, ob die Kardinalität zulässig ist.
Modifizieren Sie Ihren Entwurf nötigenfalls.

Erweitern Sie die Attribute (falls erforderlich), so daß alle geforderten Informationen Gespeichert werden können.



Zwischentabelle

Primary key

2 Relationen

Attribut „funktion“

➔ **4 PUNKTE**