

PHP'ye Giriş

PHP Nedir?

Rasmus Lerdorf tarafından öncelikle kendi kişisel web sayfalarını yazmak için geliştirilmesi nedeniyle 'P'ersonal 'H'ome 'P'ages adının kısaltması olarak karşımıza çıkan PHP, HTML gömülü (HTML-embedded) bir script dilidir. özellikle web uygulamaları geliştirenlerin dinamik ve havada (on the fly) üretilen web sayfalarını hızlı bir şekilde yazmaları için düşünülmüştür. Dilin sentaksı geliştirilirken C, Java ve en çok da Perl'den esinlenilmiştir. Bu nedenle bu dillerden her hangi birisini bilen bir kullanıcı için PHP'ye geçmek hiç de zor olmamaktadır.

PHP kendi yorumlayıcısı ile birlikte kullanılan bir script dilidir. Bu yorumlayıcının yaptığı aslında HTML kodları arasına yerleştirilmiş PHP kodlarını ayrıştırıp işleyerek sonuçta saf HTML formatında bir çıktı üretmek web sunucuya iletmektir.

Dilin Yapısı

PHP, HTML gömülü bir dildir. Yani aynı dosya içinde hem HTML, hem de PHP kodu kullanılabilir. Kod içinde HTML'den PHP'ye geçmek için bazı ayırıcılar kullanılması gerekir. Örneğin:

```
<TABLE SIZE="500">  
  <TR>  
    <TD><? Buraya PHP kodu yazılacak ?><TD>  
    ...
```

HTML tagleri içinde de PHP kullanmak için şu metod seçilmeli

```
<TABLE SIZE="<?PHP Buraha PHP kodu yazılacak ?>" ...
```

Veya:

```
<script language="php">
    Buraya PHP kodu yazılacak
</script>
```

Ve ilk 'Hello World' cümlemiz:

```
<html>
<head>
    <title>Example</title>
</head>
<body>
    <? echo "Hello World..."; ?>
</body>
</html>
```

Browser bunu bize:

```
Hello World...
```

şeklinde gösterecektir.

Değişkenler

PHP, değer ile atama (assign with value) ile birlikte adres ile atama (assign with reference) metodunu değişkenler üzerinde kullanabilir. Burada '&' işareti, önüne geçtiği değişkenin değerinin değil, adresinin işlemde kullanılmasını sağlar.

Önceden tanımlı değişkenler

PHP programcıya global olarak önceden tanımlı değişkenler sunmaktadır. Bunlar, PHP modül

olarak çalıştırılıyorsa web sunucusunun gördüğü ve PHP scriptine aktardığı çevre değişkenleri, CGI değişkenleri ve sunucunun kendi ürettiği değişkenlerdir. PHP CGI olarak çalıştırılıyorsa CGI değişkenleri hariç diğerleri değişiklik gösterebilir. PHP bir başka program içinden veya komut satırından çalıştırılıyorsa web sunucunun kendi değişkenlerini görmesi zaten sözkonusu olamayacağı gibi çevre değişkeni olarak da içinde bulunulan shell veya programın set ettiği değişkenler geçerlidir. Bunların listesi phpinfo fonksiyonu ile görülebilir.

Kullanıcıdan PHP scriptine gelen değişkenler

PHP'nin en güzel özelliklerinden birisi herhangi bir çözümleme (parsing) işlemi gerektirmeden dışarıdan gelen değişkenlerin çok rahat kullanılabilmesi:

```
<form action="foo.php3" method="post">
  Adınız: <input type="text" name="name"><br>
  <input type="submit" name="submit" value="Tamam">
</form>
```

Yukarıdaki formdan gönderilen değişkenler PHP içinde aşağıdaki gibi doğrudan kullanılabilir:

```
<? echo "Adınız:". $name; ?>
```

Variable variables

Adı dinamik değişebilen değişkenler (variable variables) programlamayı oldukça esnekleştiriyor. Yalnızca recursive yapılabileceğini düşündüğünüz pek çok işi bunlara yaptırabilirsiniz. Örneğin aşağıdaki kod çıktı olarak 'hello world' basar:

```
$a = "hello";
$$a = "world";
echo "$a $a{$a}";
```

Type Juggling & Type Casting

PHP, integer, float, string, array gibi çeşitli değişken tipleri içerir. Ancak değişkelerinizin tipi sabit değildir. Yani çalışma anında değişkeninizin tipini değiştirebilirsiniz (type juggling).

İstenmeyen durumlar oluşmaması için type casting kullanılarak da değişkenin değerinin nasıl yorumlanmasını istediğimizi belirtebiliriz.

Diziler

Dizilere rastgele (random) erişim yanında sıradan (sequential) erişim ve kuyruk (que) veri yapısı içinde kullanma özellikleri kazandırılmıştır, PHP'de, Perl'de olduğu gibi associative diziler tanımlanabilir. Bu dizilere (random) erişim yanında sıradan erişim olanağı veren list ve each gibi fonksiyonlar da tanımlanmıştır, Diziler çalışma anında büyüyebilir. (Bu özellik PHP yorumlayıcı konfigürasyonunda gerekli ayar yapılmadan dikkatsiz kullanılırsa işletim sisteminin bile göçmesine sebep olabilir). Dizilerin index boyutları da dinamik olarak değiştirilebilir.

Fonksiyonlar

- Recursive fonksiyon çağrıları yapılabilir.
- Adı dinamik değişen fonksiyonlar (variable functions) kullanılabilir.
- Static değişkenler kullanılarak fonksiyonlara geri dönüşlerde değişkenin son aldığı değer kullanılabilir.
- Fonksiyon parametreleri standart değer alabilecek şekilde belirtilebilir.
- Fonksiyonlar değişen sayıda parametre kullanabilir.

- Fonksiyon dışındaki global deęişkenlerin kullanımı için tanımlama işi programcıya bırakılmıştır.

Ayrıca eval fonksiyonu kullanılarak kod içerięi dinamik olarak oluşturulabilir:

```
eval( 'echo "Merhaba";' );
```

Modülerlik için include ve require fonksiyonları kullanılabilir.

Dosya upload etme (file uploads) ve HTTP bağlantılarını ele alma (connection handling) gibi gelişmiş özellikler de PHP ile beraber geliyor.

PHP ve Kurallı İfadeler

Kurallı İfadeler (Regular Expressions - regexp)

Kurallı ifadeler öteden beri Unix sistemlerinin vazgeçilmez unsuru olmuştur. Karmaşık metin arama, deęiştirme ve çözümleme (text processing and parsing) işlemleri için satırlar dolusu kod yazmak yerine özel semboller ile anlamlı hale getirilmiş tek satırlık bir kurallı ifade kullanılarak aynı sonuç elde edilebilir. Unix kendi sistemi içinde regexp desteęi ile gelmekte ve kabuk programları bunları entegre olmuş bir şekilde kullanabilmektedir. Aynı şekilde sed, awk, perl gibi araçlar da regexp kullanımı için örnek verilebilir.

PHP ile Kurallı İfadeler

PHP, kurallı ifadeleri iki farklı şekilde ele almaktadır.

Perl uyumlu kurallı ifadeler klasik regex formatını kullanırlar:

- Bu tür kurallı ifadeler iki tane "/" işareti arasında gösterilirler.
- Karakterler ve karakterlerden oluşan bloklar kullanılır.
- Bloklar, "()" imleçleri arasında gösterilirler.
- "+" işleci sonuna geldiği karakter veya bloğun en az bir kere metin içinde geçtiğini anlatır.
- "*" işleci sonuna geldiği karakter veya bloğun metin içinde sıfır veya daha çok kez geçtiğini anlatır.
- "?" işleci sonuna geldiği karakter veya bloğun sıfır veya bir kere metin içinde geçtiğini anlatır.
- "{num1,num2}" türü bir işleç sonuna geldiği karakter veya bloğun kaç kere metin içinde geçtiğini alt ve üst sınırlar vererek anlatır.
- "?" işleci sonuna geldiği karakter veya bloğun sıfır veya bir kere metin içinde geçtiğini anlatır.
- "." newline ("\n") dışında tüm karakterleri ifade eder.
- "\s", tab ('\t') ve newline ('\n') dahil olmak üzere white space karakterlerini ifade eder.
- "\S", white space karakterler dışındaki tüm karakterleri ifade eder.
- "\d" 0-9 arası rakamları ifade eder
- "\w" harf, rakam ve '_' gibi bütün bir kelimeyi oluşturan karakterleri ifade eder.

- "\W", "\w" nin dışındaki karakterleri ifade eder.
- "^" cümle başlangıcını ifade eder
- "\$" cümle bitişini ifade eder
- "\b" kelime başlangıç ve bitişleri için kullanılır.
- "[]" imleçleri arasında rakam veya harf aralığı içinde kalan bir karakter ifade edilir.
- "^" imleci, "[]" imleçleri arasında kullanılırsa seçimi tersine çevirir.
- "|", 'OR' içleci gibi çalışır.
- "\", tüm bu özel karakterleri olduğu gibi yorumlanmasını sağlar.

SSL, PHP ve MySQL destekli Apache kurulumu örneği

Platform:

Solaris2.6 / 7

Apache'nin kullanacağı araçlar:

- PHP (PHP-4.0) <http://www.php.net>
- OpenSSL <http://www.openssl.org>
- ModSSL (mod_ssl-2.6.4-1.3.12) <http://www.modssl.org>

Apache'de aktif edilen modüller:

- mod_so: DSO desteği
- mod_ssl: SSL desteği

PHP'nin kullanacağı araçlar:

- MySQL (mysql-3.22.27) (MySQL veritabanı - seçimlik) <http://www.mysql.net>
- OpenLDAP Client Library (LDAP desteği - seçimlik) <http://www.openldap.org/>
- IMAP C-Client Library(UWImap4.7c - içinde pop3 sunucusu da gelmekte...) (IMAP desteği - seçimlik)
 - <http://www.imap.org/>
 - <http://www.washington.edu/imap/>
 - <ftp://ftp.cac.washington.edu/imap/>
- GD Image Library (gd-1.8.2) (Grafik desteği - seçimlik) <http://www.boutell.com/gd/>
- PDF Library (pdflib-3.0) (PDF üretme - seçimlik) <http://www.pdflib.com/pdflib/>
- Zend Optimizer (Hızlandırıcı - seçimlik) <http://www.zend.com/>

GD Image Library için gerekenler:

- zlib (zlib-1.1.3)
 - <http://www.info-zip.org/pub/infozip/zlib/>

- <http://www.freesoftware.com/pub/infozip/zlib/>
- PNG Library (libpng-1.0.6)
 - <http://www.libpng.org/pub/png/>
 - <http://www.libpng.org/pub/png/pngcode.html>
 - <http://www.libpng.org/pub/png/libpng.html>
- JPEG Library (libjpeg-6b ve üzeri) <ftp://ftp.uu.net/graphics/jpeg/>
- FreeType Library (freetype-1.3.1 - True Type Fontlar için opsiyonel)
<http://www.freetype.org>

PDF Library için gerekenler:

- JPEG Library (libjpeg-6b ve üzeri) <ftp://ftp.uu.net/graphics/jpeg/>
- FreeType Library (freetype-1.3.1 - True Type Fontlar için opsiyonel)
<http://www.freetype.org>

Kurulum

Burada paketlerin kurulum sırası önemlidir.

Apache Kurulumu (DSO ve SSL desteği ile)

- OpenSSL (openssl-0.9.4)

```
sh config \
-fPIC
```

```
make
make test
make install
```

- **ModSSL (mod_ssl-2.6.4-1.3.12)**

```
configure --with-apache=../apache-1.3.12 (Apache source dizini)
```

- **Apache (apache-1.3.12) (DSO ve SSL desteği ile)**

```
SSL_BASE=../openssl-0.9.4           \ (OpenSSL source dizini veya
                                     \ OpenSSL install edilmiş ise
                                     \ SSL_BASE=SYSTEM)
configure --enable-rule=SHARED_CORE \ (DSO desteği)
        --enable-module=so          \ (DSO desteği)
        --enable-module=ssl         \ (SSL desteği)
        --enable-shared=ssl         \ (SSL, DSO olarak yüklenecek)
```

```
make
make certificate
make install
```

PHP Kurulumu (Opsiyonel Kütüphaneler ile DSO modülü olarak)

- **OpenLDAP client library**

```
configure --with-threads=lwp \
        --disable-slapd      \
        --disable-ldapd     \
        --disable-slurpd

make
make install
```

- **UW Imap 4.7c Client library**

```
make gso
```

```
cp -i c-client/c-client.a /usr/local/lib/libc-client.a
```

```
cp -i c-client/rfc822.h /usr/local/include
```

```
cp -i c-client/mail.h /usr/local/include
```

```
cp -i c-client/linkage.h /usr/local/include
```

- **zlib library**

```
configure
```

```
make
```

```
make install
```

```
make distclean
```

```
configure --shared
```

```
make
```

```
make install
```

- **png library (Burada 'Makefile' dosyası içinde zlib'in lib ve include dizinleri belirtilecek.)**

```
make
```

```
make install
```

- **jpeg library**

```
configure
```

```
make
```

```
make install
```

```
make install-lib
```

```
make distclean
```

```
configure --enable-shared
```

```
make
```

```
make install
```

```
make install-lib
```

- **freetype library**

```
configure
```

```
make
```

```
make install
```

- **gd library** Öncelikle Makefile üzerinde değişiklikler yapılmalı. Örneğin JPEG ve TTF desteği açılmalı,include dizinleri eklenmeli vs...

```
make
```

```
make install
```

- **pdf library**

```
configure
```

```
make
```

```
make install
```

- **mhash library**

```
configure
```

```
make
```

```
make install
```

```
configure \
    --disable-debug \ (Zend Optimizer için)
    --with-apxs=/apache-1.3.12/bin/apxs \ (Apache'nin kurulduğu dizin.
    \ DSO modülü olarak Apache'ye ekleme)
    --enable-track-vars \ (HTTP_POST_VARS, HTTP_GET_VARS,
```

```
        \ HTTP_COOKIE_VARS gibi global associative
        \ array tanımlamalarını etkinleştirir)

--enable-ftp        \ (FTP desteği)
--enable-sysvsem    \ (System V semafor desteği)
--enable-sysvshm    \ (System V shared memory desteği)

--with-mysql        \ (MySQL client ve fonksiyonları)
--with-ldap         \ (LDAP fonksiyonları)
--with-imap         \ (IMAP fonksiyonları)
--with-mcrypt       \ (Encryption fonksiyonları)
--with-mhash        \ (Hash fonksiyonları)
--with-zlib         \ (GD ve PDF için compression)
--with-ttf          \ (PDF için opsiyonel)
--with-gd           \ (Image fonksiyonları için)
--with-pdflib       \ (PDF fonksiyonları için)
--with-zlib-dir     \ (GD ve PDF için gerekli)
--with-jpeg-dir     \ (GD ve PDF için gerekli)
--with-png-dir      \ (GD için gerekli)
```

```
make
```

```
make install
```

'httpd.conf' dosyasına şu satırlar otomatik eklenmiş olmalı

```
LoadModule  php4_module  libexec/libphp4.so
AddModule   mod_php4.c
```

Zend Optimizer

```
mkdir /usr/local/Zend
```

```
mkdir /usr/local/Zend/lib
```

```
cp ZendOptimizer.so /usr/local/Zend/lib
```

'php.ini' dosyasına şu satırlar eklendi:

```
zend_optimizer.optimization_level=7
```

```
zend_extension="/usr/local/Zend/lib/ZendOptimizer.so"
```

PHP ile Doğrulama (Authentication) ve Güvenlik

Çok kullanıcı sistemlerin tümünde güvenlik (security), önemli bir sorun olmuştur. Web uygulamalarında da verilerin ve kaynakların yetkisiz kişilerin erişim ve kullanımına kapatılması gerekebilir. Dolayısıyla bir Web uygulaması gerçekleştiriminde de güvenliğe dikkat edilmesi gerekmektedir.

Güvenlik, birkaç farklı kavramı içerisine alır. Bunlar arasında doğrulama (authentication), yetkilendirme (authorization) ve gizlilik (privacy) sayılabilir.

Bir Web-mail uygulamasında posta kutusu mesajları ile IMAP ve sendmail kaynaklarına sadece ilgili posta kutusu sahibinin erişebilmesine izin verilmelidir. Burada Web sunucu ile posta kutusu sahibi arasında bir diyalog gerçekleşecek ve her iki taraf da birbirine kendisini tanıtacaktır.

Bunlardan kullanıcı kendisinin söylediği kişi olduğuna delil olacak bilgiyi karşı tarafa iletecek ve karşı taraf doğrulama sağlarsa oturumu açacaktır. İşte burada örneklenen, doğrulama (authentication) sürecidir.

Bir veritabanından sorgulama yapabilecek kişiler ile veritabanında değişiklik yapabilecek

kişilerin yetkilerinin web uygulaması tarafından doğru bir şekilde ele alınabilmesi gerekmektedir.

Burada ise yetkilendirme (authorization) sorunu ile karşılaşmaktayız.

Web-mail uygulamasına dönecek olursak, kullanıcının postalarını tarayıcısına (browser) çekerken üçüncü bir kişinin araya girip bunları görebileceği açıktır. Burada ise gizlilik (privacy) ön plana çıkmakta ve bu sorun da daha çok çeşitli şifreleme (encryption) metodları ile çözülmeye çalışılmaktadır.

PHP ile Oturum Yönetimi (Session Management)

HTTP protokolünün o anki durumu kaydetmeyen (stateless) yapısı web uygulamaları için en önemli sorunu oluşturmuş ve bunun aşılabilmesi için hem sunucu, hem de istemci tarafında çeşitli metodlar geliştirilmiştir.

Web sunucusuna bağlanan bir kişi web tarayıcısı aracılığıyla sunucudan bir URL adresi ile belirtilmiş kaynağı ister (HTTP request) ve sunucu da istemi değerlendirir ve sunucu tarafında yapılması gerekenleri yaparak tarayıcıya cevabını iletir (HTTP response). Arada sürekli bir TCP/IP bağlantısı yoktur. Aslında kaynak kullanımı ve bant genişliği açısından oldukça avantajlı olan bu durum web uygulaması geliştirirken dezavantaja dönüşür. Bir web uygulamasında web sunucusuna çeşitli istemlerde bulunan kişi, bu istemleri arasında o an hangi aşamaya gelmişse oraya kadar yaptığı işlemlerin kaybedilmemesini, sunucu tarafından hatırlanmasını isteyecektir. Bu sorun sunucu tarafta oturum yönetimi (session management) kavramının doğmasına sebep olmuştur.

Web üzerinde oturum yönetimi sunucu ve istemci tarafta olmak üzere ikiye ayrılabilir:

İstemci tarafta oturum yönetimi:

İstemci tarafta oturum yönetimi kavramını, bir HTTP oturumu esnasında istemci tarafa bir önceki oturuma ilişkin bilgiler atmak veya istemci taraftan bir önceki oturuma ilişkin bilgileri almak şeklinde düşünülebilir.

Eğer HTML'in form olanağından yararlanılıyorsa HIDDEN türü alanlar kullanılarak bir önceki oturum bilgisinin istemciye aktarılması ve tekrar form aracılığı ile sunucuya geri iletilmesi sağlanabilir. Uygulamanız formlardan oluşuyor ise oldukça kullanışlıdır.

Bir başka metod oturum bilgilerini URL içine gömme metodudur. Form kullanılmasına gerek kalmadan HTML linkleri içine GET metodu çağrılarının yaptığı gibi gömerek uygulama ekranları arasında bilgi taşınabilir.

Son olarak cookie'ler kullanılabilir. Cookie'lerin çalışma mantığı basittir. Browser web sunucusuna istemde bulunurken gönderdiği header bilgisinin arkasına daha önceden ilgili sunucunun set ettiği cookie değişkenleri ve bunların değerlerini gönderir. Sunucu bu bilgileri değerlendirerek en son nerede kaldığını hatırlayabilir. Daha sonra sunucu, browsere cevap gönderirken set etmek istediği cookie değişkenlerini ve değerlerini cevap iletisinin header kısmına koyar ve browser yoksa yeni cookie değişkenleri yaratır ya da varsa önceki cookie değişkenlerinin üstüne yazar.

Cookie'ler isim, değer, zaman aşımı süresi, kaynak sunucu domain adresi ve istenen URI path bilgisi gibi alanlar içerebilirler.

İstemcilere cookie göndermek için setcookie fonksiyonu kullanılır.

İstemcilerden gelen cookieler ise \$HTTP_COOKIE_VARS global associative array'inden cookie adı index olarak verilerek okunabilir.

Cookieleri sisteminizde tanımlı kullanıcıları ayırdetmekte kullanabilirsiniz. LDAP sunucusu bağlantısı için kullanıcı adı ve şifresini tekrar tekrar girmemesini sağlamak için ilk olarak login ekranını gönderen script içinde komutları ile browserlerindeki user ve pwd cookieleri temizleniyor:

```
setcookie("user");  
setcookie("pwd");
```

Sonra formdan girilen değerler ikinci bir scriptde browsere set edilir:

```
setcookie("user", $user);  
setcookie("pwd" , $pwd);
```

Doğrulamanın yapılacağı scriptte bu değerler alınıyor:

```
$user = $HTTP_COOKIE_VARS["user"];  
$pwd = $HTTP_COOKIE_VARS["pwd"];
```

Bu metod çalışmakla birlikte yalnızca güvenli olduğunu düşündüğünüz bir bölgede kullanılmalıdır. Aksi taktirde HTTP üzerinden açık bir şekilde giden kullanıcı adı ve şifresi rahatça alınabilir. Bu nedenle authentication gereken web bölgeleri ve uygulamaları için secure socket layer kullanılması (HTTPS protokolü üzerinden) önerilir.

Setcookie fonksiyonu zaman aşımı süresi vermeden kullanılırsa cookie ömrü web tarayıcısının ömrü ile sınırlıdır (browser tarafından cache edilir). Zaman aşımı süresi verilirse cookie bilgisi dosyaya yazılır ve browser kapatılsa da zaman aşımı süresi dolana kadar silinmez.

Cookielerinizin başka domainler ve hatta URI'ler tarafından çekilebilmesini engellemek için

domain name ve path parametreleri de kullanılabilir.

Ayrıca setcookie fonksiyonuyla sadece HTTPS kullanılıyorsa cookie gönderilmesi sağlanabilir.

Tüm bu metodlarda oturum verisini taşıırken HTML'e özgü karakterleri HTML yorumlayıcısının ihmal etmesini sağlamak amacıyla HTTP entitylerine otomatik çevirme veya URL encoding-decoding işlemleri yapmak gerekebilir. Cookie'ler gönderilip alınırken encoding-decoding işlemleri otomatik yapılır. Ayrıca serialize ve unserialize fonksiyonları kullanılarak programınız içindeki karmaşık veri yapısında tuttuğunuz değişkenleri byte dizisi haline getirip istemci tarafa atmak ve alıp tekrar orjinal haline çevirmek çok kolaydır.

Sunucu tarafta oturum yönetimi:

NSEnterprise Server'deki session objectlerini karşılayacak yapılar shared memory ve semafor kullanılarak kurulabilir. Hafızada script ömrüyle sınırlı olmayan alanlar ayırmak için shared memory fonksiyonları kullanılabilir. Aynı scriptin hafızadaki tüm kopyalarının okuyabileceği paylaşılabilir bir alan yaratılır:

```
define(_SHMKEY,1000); // Kendi atadığımız bir değer. Biricik olmalı
define(_SHMSIZE,1000); // 1000 Bytelık bir alan
define(_SHMPERM,0600); // Yalnızca web sunucunun uid'sine
                        // sahip kullanıcı okuyup yazabilsin
$shm_id = shm_attach(_SHMKEY,_SHMSIZE,_SHMPERM);
```

Artık shared memoy'mizden okumak için:

```
$value = shm_get_var($shm_id, $index);
```

Shared memory'e yazmak için:

```
shm_put_var($shm_id, $index, $value);
```

kullanılabilir. Burada '\$index' tamsayı olmak zorudur. Fakat shared memor'ye bu şekilde erişmek güvenli olamayacağı için semafor kullanımı şarttır:

```
define(_SEMKEY,999); // Kendi atadığımız bir değer. Biricik olmalı
define(_SEMPERM,0600); // Yalnızca web sunucunun uid'sine
// sahip kullanıcı bu semaforu kullanabilsin
$sem_id = sem_get(_SEMKEY, 1, _SEMPERM);
```

Şimdi shared memory işlemlerinin yapılacağı kritik kesim semafor işlemleri arasına alınarak sözkonusu hafıza bölgesi güvenli bir şekilde kullanılabilir:

```
sem_acquire($sem_id); //--- Semafor aynı scriptin başka bir kopyası
//--- tarafından alınmışsa burada bekler
```

```
{ Kritik kesim }
```

```
sem_release($sem_id);
```

Burada dikkat edilmesi gereken nokta web sunucunun uid'i ile çalışan herhangi bir görevin shared memory alanına erişebileceği riskidir.

Hibrit Yaklaşım ve PHP Session Desteği:

Oturum yönetiminde hem sunucu, hem de istemci tarafı bir arada kullanmak gerekebilir. Dış dünyaya açık bir sistemde bağlanan tüm kullanıcıları sistemin o an biricik bir kullanıcı olarak tanıması ve izleyen oturumlarda da bu kullanıcıyı hatırlaması istenebilir. Aynı şekilde kullanıcı doğrulaması için gerekli bilgilerin istemci tarafa aktarımının riskli olduğu açıktır. Bu sorunları aşabilmek için kullanıcılara biricik tanımlayıcılar atanarak, istemci tarafa sadece id aktarımı yaparak, verilerin yüklü bir kısmını bu id'lerle erişilebilir şekilde sunucu tarafta tutmak tercih

edilir.

Burada kritik nokta farklı kullanıcıları birbirinden ayırmak için biricikliği sağlamak ve bu ID'leri tahmin edilemez şekilde üretmektir.

```
$ID=uniqid(getenv("REMOTE_HOST"));  
setcookie("id",$ID,time()+3600,"/",".ulakbim.gov.tr");
```

Yukarıdaki kısım çalıştırılarak kullanıcı tarayıcısına biricik bir ID atanarak bu ID'ye ait oturum verilerinin geçici dosya, shared memory ve veri tabanında tutulması sağlanabilir.

Bir kez kullanıcı doğrulaması yapıldıktan sonra böyle bir cookie kullanılarak kullanıcı adı ve şifresi aktarılmadan sadece üretilen biricik ID değeri (token) aktarılarak doğrulamalara devam etmek mantıklı olacaktır. Burada token'in tahmin edilemeyecek ve çakışma olmayacak bir şekilde üretilmesi önem taşır. Bu amaçla md5 digestleri üretmek de tercih edilebilir.

PHP'nin 4.0 versiyonuyla gelen session desteği, cookie veya URL'e gömerek session ID'si aktarımı ve sunucu tarafta oturum verilerinin tutulması mantığını programcıya saydam hale getiriyor.

session_start fonksiyonu, istemci taraf için, yok ise bir session ID yaratarak cookie olarak set eder.

session_register fonksiyonu, başlatılan oturum için değişkenler üretir. Aşağıdaki kod değişkeninizin oturumlar arasında sürekli artmasını sağlıyor:

```
session_start();  
session_register($count);  
  
if(!isset($count) || !isset($var1))
```

```
$count=0;  
else  
    $count++;  
  
echo $count;
```

PHP geliřtiriciler arasında popöler olan PHPLIB base library geliřmiř oturma yönetimi deiteđi sađlamaktadır.

MySQL Veritabanı Yönetim Sistemi ve SQL

MySQL

MySQL, çok-kanallı (multi-threaded), çok kullanılcılı (multi-user), hızlı ve sađlam (robust) bir veritabanı yönetim sistemidir. Kaynak kodu açık olan MySQL'in pekçok platform için binaryleri de hazır bulunmaktadır. ODBC sürücöleri de mevcuttur. Çok sayıda tablo ve kayıttan oluřan büyük ölçekli veri tabanları oluřturularak kolayca yönetilebilmesine imkan sađlanmıřtır.

MySQL'in kendi sitesine ařađıgaki adresten ulařılabilir:

- <http://www.mysql.org>

MySQL veritabanı yönetim sistemini hiyerarřik olarak řu sırada düřünebiliriz:

- Veritabanı (database)
- Tablo (table)

- Kayıt (record)
- Veri tipi (data type / column type)

Veri Tipleri:

- char
- varchar
- int
- float
- date
- text/blob
- set
- enum

Tablolar:

- create table
- alter table
- drop table
- create index

Veritabanları

- create database
- drop database

Veri Tanımlama ve Değişirme (Data Definition and Manipulation):

- select
- insert
- update
- delete

Erişim Hakları (Privileges):

MySQL erişim haklarını mysql adlı özel bir veritabanında tutar. Bu veritabanı birtakım erişim hakkı tablolarından oluşur (grant tables):

user tablosu kullanıcıların sunucuya hangi host makinadan hangi haklarla bağlanabileceğini belirler. Bu tabloda bulunan haklar kullanıcıya global olarak verilir ve verilen haklar istisnasız tüm veritabanları üzerinde etkili olur. Host alanı ise tam tersine bağlanabilecek tüm hostları içermelidir. Diğer tablolarda bunların dışında host alanı girilmesi anlam taşımayacaktır.

db tablosu kullanıcıların belirli bir veritabanına hangi haklarla erişebileceğini gösterir. Global olarak sahip olunmayan hakları verebilir, ancak global olarak sahip olunan hakları kaldıramaz. Yani user tablosunda kullanıcıya verilen hakları geri alamaz ama belirli bir veritabanına ait olmak üzere yeni haklar verebilir. Burada host alanı user tablosunda verilen host aralığını daraltabilir.

Yani bir kullanıcı user tablosundaki host aralığının kapsadığı bir host'tan sunucu bağlantısı kursa da, db tablosunda belirli bir veritabanına kullanıcının bu hosttan erişimine ilişkin bir satır (kayıt) yoksa veritabanına erişemez. Burada host alanı için için tipik olarak '%' kullanılabilir. Böylece sunucu bağlantısı sağlayabilen tüm hostlar veritabanına kabul edilirler.

host tablosu db tablosundaki 'host' alanına değer girilmezse anlam kazanır. Bu durumda ilgili veritabanına bağlanabilecek hostlar host tablosundan taranır ve erişim haklarına bakılır. Burada haklar açısından user ve db tabloları arasındaki ilişkiden farklı bir durum vardır. Öyle ki burada db ve host tabloları arasında adeta bir AND işlemi yapılır.

table-priv tablosu tablo düzeyinde erişim haklarını belirler. (Sadece grant, revoke komutları ile değiştirilmeli) column-priv tablosu tablo kolonları düzeyinde erişim haklarını belirler. (Sadece grant, revoke komutları ile değiştirilmeli) Bu tablolarda kullanıcının bir erişim hakkı varsa daha alt düzey tablolara bakılmaz. Yapılan değişiklikleri etkinleştirmek için flush privileges komutu kullanılır.

Veri değiştirme komutları ile mysql veritabanı üzerinde değişiklik yaparak kullanıcı hakları değiştirilebileceği gibi daha emniyetli metod olarak grant ve revoke komutları kullanılabilir.

PHP ile MySQL'in Kullanılması

PHP 4.0'dan önceki versiyonlarında MySQL fonksiyonlarını kullanabilmek için MySQL paketi ile birlikte gelen API kütüphanesine gerek duyuyordu. PHP 4.0 versiyonu MySQL API'si ile birlikte geliyor.

Adımlar:

PHP ile MySQL kullanırken tipik olarak Őu adımlardan geilir:

Veritabanı Sunucusuna Baėlanma

- *mysql_connect fonksiyonu*, verilen TCP/IP adresinde alıŐan veritabanı sunucusuna belirli bir veritabanı kullanıcısı kimliėi ile baėlantı kurar ve bir baėlantı numarası (link identifier) geri dndürür. Baėlantı ömrü script ömrü ile sınırlıdır.
- *mysql_pconnect fonksiyonu*, *mysql_connect* fonksiyonu ile aynı işlevi veritabanı sunucusuna sürekli (persistent) bir baėlantı kurarak yapar. Baėlantı ömrü, Web sunucusunun ömrüyle veya *mysql_close* fonksiyonu ile özellikle (explicitly) belirtilmesiyle sınırlıdır.

Veritabanı Seçimi

- *mysql_select_db* fonksiyonu, verilen veritabanını o anki aktif veritabanı haline getirir (seilen veritabanı üzerine konumlanılır). Sunucuya baėlanırken verilmiş kullanıcı kimliėinin bu veritabanına erişim hakkı yok ise bu konumlanma gerçekleşmez. Aktif bir veritabanı seçimi istenmiyorsa bu aşama atlanabilir.

Sorgu Yönelme (Queries)

Alınan baėlantı numarası kullanılarak sunucuya bir sorgu yöneltilir ve bir sonuç numarası (handle) alınır.

- *mysql_query fonksiyonu*, o anki aktif veritabanı üzerinde sorgu işletir.

- `mysql_db_query` fonksiyonu, parametre olarak verilen veritabanına konumlanarak üzerinde sorgu iřletir.

Sorgu Sonuları (Query Results)

Numarası ile tanımlı sonuç nesnesinden eřitli sonuçlar PHP ile kullanılabilir őkilde deėiřkenler üzerine alınabilir:

- `mysql_num_rows` fonksiyonu, sonuç numarası verilen sorgu sonucunda dndrlen kayıt sayısını verir.
- `mysql_fetch_row` fonksiyonu, sonuç numarası verilen sorgu sonucunda dndrlen kayıtları bir dizi (array) olarak geri dndrr.
- `mysql_fetch_array` fonksiyonu, sonuç numarası verilen sorgu sonucunda dndrlen kayıtları associative bir dizi (array) olarak geri dndrr.

Sunucu Baėlantısını Keme

- `mysql_close` fonksiyonu, numarası verilen veritabanı sunucu baėlantısını keser. Script iřletimi sonlanınca baėlantı otomatik olarak (implicitly) kesilir.

rnek Uygulamalar

Telefon Rehberi

Telefon rehberi uygulaması iin ilk olarak kullanılacak veritabanı tabloları SQL ifadeleri olarak

tanımlanmalı. Önce veritabanı sunucusuna bağlantı kuruluyor:

```
mysql -u root -p
```

Veritabanı yaratılıyor:

```
mysql> CREATE DATABASE rehber;
```

Veritabanı seçiliyor:

```
mysql> USE rehber;
```

Tablolar yaratılıyor:

```
mysql> CREATE TABLE personel (  
    -> personel_ID SMALLINT NOT NULL AUTO_INCREMENT,  
    -> adsoyad VARCHAR(50) NOT NULL,  
    -> telefon VARCHAR(20),  
    -> odano VARCHAR(5),  
    -> email VARCHAR(50),  
    -> evtel VARCHAR(20),  
    -> ceptel VARCHAR(20),  
    -> bolum_ID SMALLINT,  
    -> KEY (personel_ID)  
    -> );
```

```
mysql> CREATE TABLE bolum (  
    -> bolum_ID SMALLINT NOT NULL AUTO_INCREMENT,  
    -> bolum VARCHAR(50) NOT NULL,  
    -> birim_ID TINYINT,  
    -> KEY (bolum_ID)  
    -> );
```

```
mysql> CREATE TABLE birim (  
    ->     birim_ID TINYINT NOT NULL AUTO_INCREMENT,  
    ->     birim VARCHAR(50) NOT NULL,  
    ->     KEY (birim_ID)  
    -> );
```

Bu SQL komutları bir dosyaya kaydedilirse mysql istemcisine standart input olarak girilebilir.

Örneğin komutlar tables.sql dosyası olarak kaydedildikten sonra komut satırından işletilebilir:

```
mysql -u root -p < tables.sql
```

İkinci olarak veritabanı üzerindeki roller tespit ediliyor:

- Normal kullanıcı, veritabanı üzerinde sorgu yapabilmelidir.
- Yetkili kullanıcı (admin user), normal kullanıcıların haklarına sahip olduğu gibi veritabanı üzerinde değişiklik (ekleme, silme, güncleme) yapabilmelidir.

Veritabanı seçiliyor:

```
mysql> use mysql
```

Yetkili kullanıcı yaratılıyor:

```
mysql> INSERT INTO user (host, user, password) VALUES (  
    ->     'localhost',  
    ->     'rehadmin',  
    ->     password('rehadmin')  
    -> );
```

Burada şunlar gerçekleşiyor:

- Sadece sunucunun bulunduğu makineden ('localhost') bağlanılabileceği belirtildi.
- 'rehber' isimli bir kullanıcı yaratıldı.

- Kullanıcının erişim şifresi encrypt edilerek veritabanına kaydedildi.
- Burada verilen haklar sunucumuzun baktığı tüm veritabanlarına erişimi ilgilendirdiğinden güvenlik için hiçbir hak verilmedi.

Normal kullanıcı yaratılıyor:

```
mysql> INSERT INTO user (host, user, password) VALUES (  
-> 'localhost',  
-> 'reuser',  
-> password('reuser')  
-> );
```

Burada da kullanıcıya verilen haklar sunucumuzun baktığı tüm veritabanlarına erişimi ilgilendirdiğinden güvenlik için hiçbir hak verilmedi.

Kullanıcıların veritabanına erişim hakları düzenleniyor:

```
mysql> INSERT INTO db  
-> ( host, db, user,  
-> Select_priv, Insert_priv, Update_priv, Delete_priv,  
-> Create_priv, Drop_priv, Alter_priv)  
->  
-> VALUES (  
-> 'localhost', 'rehber', 'readmin',  
-> 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y'  
-> );
```

Yukarıda 'rehber' isimli veritabanına 'readmin' kullanıcısının sadece 'localhost'tan erişebileceği ve yalnız 'select', 'insert', 'update', 'delete', 'tablo yaratma' ve 'tablo silme' hakkı olduğu

belirtiliyor.

```
mysql> INSERT INTO db
->   ( host, db, user, Select_priv)
->
->   VALUES(
->     'localhost', 'rehber', 'rehuser', 'Y'
->   );
```

Yukarıda 'rehber' isimli veritabanına 'rehuser' kullanıcısının sadece 'localhost'tan erişebileceği ve yalnız 'select' hakkı olduğu belirtiliyor.

MySQL sunucusu çalışmaya başlarken erişim tablolarını hafızaya yükler ve erişim yapılacağı zaman tablolardan okumak yerine cachine bakar. Dolayısıyla yaptığımız bu değişikliklerin etkin olabilmesi için cache'in yeniden yüklenmesi gerekir:

```
mysql> FLUSH PRIVILEGES;
```

Bu komutu da ancak reload hakkı olan kullanıcı çalıştırabilir.

Bu SQL komutları da users.sql dosyasına kaydedildikten sonra aynı şekilde komut satırından işletilebilir:

```
mysql -u root -p < users.sql
```

Bundan sonraki aşamada veritabanına mevcut kayıtların girilmesi gerekiyordu. Başka bir mysql sunucuda olarak tutulan veritabanı verisi ihraç edilerek (dumping) elde edilen '.txt' türü metin dosyası kendi veritabanımıza dahil edildi (import). Bu noktada import edilecek metin dosyası herhangi bir veri tabanından ayrıştırılmış metin dosyası halinde alınabiliyor ise mysqlimport uygulaması ile kolayca MySQL veritabanına aktarılabilir.

Veritabanının sahibi rehber olduđu için 'rehber' kullanıcısı olarak veritabanı sunucusuna oturum açılarak ařağıdaki komutlar uygulandı:

Kullanıcı dizine ilk girişinde index.php3 (kaynak kodu) üzerinden search.php3 (kaynak kodu) sayfasına yönlendiriliyor. Burada kullanıcıya gönderilen formun da örneğın header ve footer kısımlarında standart ve dinamik olması için php kullanılıyor. Yine combo box'da da opsiyonlar veritabanından alınarak dinamik olarak oluşturuluyor.

Kullanıcı, formu submit edince srchres.php3 (kaynak kodu) çağrılır. Bu script içinde bir query cümlesi girilen alan değerlerine göre dinamik olarak oluşturulur.

PHP ile LDAP Uygulamaları

LDAP (Lightweight Directory Access Protocol), bir dizin servisi protokolü türevidir. Dizin servisleri hiyerarşik bir yapıda (dizin yapısında), veriye merkezi olarak ulaşılması için düşünölmüş bir bakıma veritabanı hizmeti veren sistemlerdir. Açık sistem olması dünya üzerinde global olarak erişilebilir bilgi kaynakları oluşturulabilmesini sağlar. Bunu yaparken coğrafi yapı (ülke, şehir), kurum içi organizasyon ve diğeri özellikleri içeren hiyerarşik bir yapı kullanır.

Özellikle kurum içi organizasyon ve personelin kayıtlarının tutulabileceğı elverişli bir ortam sunar. Bir kurumda çalışan veya bir üniversitede okuyan öğrencilere çeşitli servisler sunmanız gerekir. Bunun için her servisin üzerinde çalıştığı makinada ayrı ayrı kullanıcı hesapları açmanız gerekebilir. Böyle olunca sistemin ve kullanıcıların yönetimi zorlaşır. LDAP kullanarak bu sorun çözülebilir.

Kuruma giren personel için LDAP sunucusuna tek bir entry eklersiniz ve LDAP desteğı veren

servislerin bu sunucu üzerinden kullanıcı bilgilerine erişim ve doğrulama yapmalarını sağlayabilirsiniz.

PHP'de LDAP için iyi bir arabirim sunmaktadır. Bu sayede web uygulamalarınızda da LDAP kullanılabilmekte ve özellikle kullanıcılara kendilerine ait verileri güncelleyebilme olanağı sunuluyor.

LDAP Dizin Servisi:

LDAP dizinleme sisteminde her kayda ait özellikler (attributes) ve bu özelliklerin değerleri (values) vardır. Her kaydın ait olduğu bir nesne sınıfı (object class) vardır. Bu yapı sayesinde dünya üzerindeki her LDAP kaydı biricik olur. Bunun için DNS de olduğu gibi hiyerarşik bir isimlendirme kullanılmıştır. Kullanıcı bu biricikliğini DN (Distinguished Name) özelliğinden alır.

Örneğin dn='cn=Mustafa Hadi Dilek, o=Ulakbim, c=tr' benim Türkiye içinde, Ulakbim adlı kuruluştta, adı 'Mustafa Hadi Dilek' olan kişi olduğumu söyler ve biriciktir.

Örneğin bir kurumda çalışan personele ait bazı özellikler şunlar olabilir:

- object class=inetorgperson, entry'nin ait olduğu sınıf. Buradaki hiyerarşi top->person->orgperson->inetorgperson şeklinde düşünülebilir,
- uid=hadi (UserID/sistemdeki kullanıcı adı),
- cn=Mustafa Hadi Dilek (CommonName/kullanıcının adı soyadı),
- sn=Dilek (SurName/kullanıcının soyadı),
- mail=hadi@ulakbim.gov.tr (kullanıcının e-mail adresi),

- telephonenumber=2989365 (kullanıcının ofis telefonu),

Genellikle sistemde yeni bir kullanıcı açarken 'inetorgperson' sınıfı kullanılır.

PHP ile LDAP Kullanımı:

PHP ile LDAP'a erişim oldukça kolaydır. İzlenmesi gereken sıra:

LDAP sunucusuna bağlanma (connect)

- *ldap_connect* fonksiyonu ile LDAP sunucusuna socket bağlantısı açılır ve bir bağlantı numarası(link identifier) alınır.

LDAP sunucusuna belirli bir noktadan erişim için doğrulama (bind)

- *ldap_bind* fonksiyonu ile sunucu veritabanında tanımlı bir kullanıcı olarak bağlantı kurulur. Bundan sonra kullanıcının sahip olduğu haklar ile sınırlanmış işlemler yapılabilir. Kullanıcı, dn özelliği ile belirtilir. Eğer dn ve password alanları girilmez ise bağlantı anonymous olarak gerçekleşir.

LDAP arama ve değiştirme türü işlevleri kullanma

- *ldap_search* fonksiyonu ile verilen kriterlere (filter) göre arama yapılır. Sonuçlar bir sonuç numarası (result identifier) alınır.
- *ldap_modify* fonksiyonu ile verilen verilen entry, yeni entry ile değiştirilir. Girilen yeni entry çok boyutlu bir dizi olmalıdır.

Elde edilen sonuçları kullanma

- *ldap_get_entries* fonksiyonu ile verilen sonuç numarasının gösterdiği sonuçlar çok

boyutlu diziye alınır.

LDAP sunucusu ile bağlantıyı kesme (close connect)

- *ldap_close* fonksiyonu sunucu bağlantısını kapatır.

Örnekler:

Telefon Defteri. LDAP üzerinden personel kayıtlarını görme, arama ve değişiklik olanağı.

Kaynak kod:

index.php - Giriş scripti

```
<?
    header("Location: rehber.php");
    exit;
?>
```

login.php - Login scripti

```
<?
//-- Telefon Rehberi --
// login.php
// LDAP uzerinde degisiklik icin login ekrani uretir.
// Yazan: Mustafa Hadi Dilek, Subat 2000
//-----
    require("common.php");
    if (!ValidateHOST()) {
        CommonHeader("LDAP Sunucusuna Bağlan");
        echo "Üzgünüz. LDAP bilgileri ancak Ulakbim içindeki bilgisayarlardan
değiştirilebilir...";
```

```
CommonFooter($footer_links);

exit;

}

if(!isset($user)) $user="";

setcookie("user");

setcookie("pwd");

CommonHeader("LDAP Sunucusuna Bağlan");

//-----
?>

<FORM ACTION="modify.php" METHOD="POST" ENCTYPE="application/x-www-form-
urlencoded">

<?
//-----
echo "<TABLE WIDTH=\"".$_WIDTH.\"\" BORDER=\"0\" CELLSPACING=\"1\"
CELLPADDING=\"1\">\n";
//-----
?>

<TR ALIGN="left" VALIGN="top">
<TD CLASS="formlabel">Kullanıcı Adı:&nbsp;  </TD>

<?
//-----
echo "<TD CLASS=\"formvalue\"><INPUT NAME=\"user\" TYPE=\"Text\""
```

```
VALUE="\$user\"></INPUT></TD>";
//-----
?>

</TR>

<TR ALIGN="left" VALIGN="top">
    <TD CLASS="formlabel">Sifre:&nbsp;</TD>
    <TD CLASS="formvalue"><INPUT NAME="pwd" TYPE="Password"></INPUT></TD>
</TR>

<TR ALIGN="left" VALIGN="top">
    <TD CLASS="formlabel">&nbsp;</TD>
    <TD CLASS="formvalue"><INPUT NAME="submit" TYPE="Submit"
VALUE="Bağlan"></INPUT></TD>
</TR>
</TABLE>
</FORM>

<?
    $footer_links["rehber.php"] ="Ulakbim Telefon Rehberi";
    CommonFooter($footer_links);

    if (isset($empty)) echo "<SCRIPT>alert(\"Kullanıcı adı ve şifreyi
girmelisiniz...\");</SCRIPT>\n";

    if (isset($unauth)) echo "<SCRIPT>alert(\"Kullanıcı olarak
tanımlanamadınız...\");</SCRIPT>\n";

    if (isset($enablecookie)) echo "<SCRIPT>alert(\"Tarayıcınızda cookie'leri
etkinleştirip tekrar deneyin...\");</SCRIPT>\n";
?>
```

rehber.php - Arama ekranı içeren ana script

```
<?
//-- Telefon Rehberi --
// rehber.php
// Arama formunu ve sonuclari uretir. LDAP tabanlı.
// Turkce karakterleri siralama destegi henuz yok.
// Yazan: Mustafa Hadi Dilek, Subat 2000
//-----
require("common.php");

if(!isset($cn)) $cn="";
if(!isset($telephonenumber)) $telephonenumber="";
if(!isset($roomnumber)) $roomnumber="";
if(!isset($mail)) $mail="";

CommonHeader("Ulakbim Telefon Rehberi");

if (ValidateHOST()) $footer_links["login.php"] ="LDAP Kaydı Değişirme";
//-----
?>

<FORM ACTION="rehber.php" METHOD="post" ENCTYPE="application/x-www-form-
urlencoded">
<INPUT TYPE="hidden" NAME="query" VALUE="YES">

<!-- Eski hali
<?
//-----
```

```

echo "<TABLE WIDTH=\"\"._WIDTH.\"\" BORDER=\"0\" CELLSPACING=\"1\"
CELLPADDING=\"1\">\n";
//-----
?>
<TR>
    <TD ALIGN="right" WIDTH="100"></A><? echo $ldap_attr_str["cn"]; ?></TD>
    <TD><? echo "<INPUT TYPE=\"text\" NAME=\"cn\" VALUE=\"$cn\">"; ?></TD>
</TR>
<TR>
    <TD ALIGN="right"><? echo $ldap_attr_str["telephonenumber"]; ?></TD>
    <TD><? echo "<INPUT TYPE=\"text\" NAME=\"telephonenumber\"
VALUE=\"$telephonenumber\">"; ?></TD>
</TR>
<TR>
    <TD ALIGN="right"><? echo $ldap_attr_str["roomnumber"]; ?></TD>
    <TD><? echo "<INPUT TYPE=\"text\" NAME=\"roomnumber\"
VALUE=\"$roomnumber\">"; ?></TD>
</TR>
<TR>
    <TD ALIGN="right"><? echo $ldap_attr_str["mail"]; ?></TD>
    <TD><? echo "<INPUT TYPE=\"text\" NAME=\"mail\" VALUE=\"$mail\">"; ?></TD>
</TR>
<TR>
    <TD>&nbsp;</TD>
    <TD><INPUT TYPE="submit" NAME="Ara" VALUE="Ara"></TD>
</TR>
</TABLE>
-->

```

```

<FIELDSET>
<?
//-----
echo "<TABLE WIDTH=\"\"._WIDTH.\"\" BORDER=\"0\" CELLSPACING=\"1\"
CELLPADDING=\"1\">\n";
//-----
?>
<TR>
    <TD CLASS="formlabel">
        <label accesskey="a" for="fcn"><? echo $ldap_attr_str["cn"]; ?></label>
    </TD>
    <TD CLASS="formvalue">
        <? echo "<INPUT TYPE=\"text\" NAME=\"cn\" ID=\"fcn\" VALUE=\"$cn\">"; ?>
ve
    </TD>
</TR>
<TR>
    <TD CLASS="formlabel"><label accesskey="t" for="ftelephonenumber">
        <? echo $ldap_attr_str["telephonenumber"]; ?></label>
    </TD>
    <TD CLASS="formvalue">
        <? echo "<INPUT TYPE=\"text\" NAME=\"telephonenumber\"
ID=\"ftelephonenumber\" VALUE=\"$telephonenumber\">"; ?> ve
    </TD>
</TR>
<TR>
    <TD CLASS="formlabel"><label accesskey="o" for="froomnumber">
        <? echo $ldap_attr_str["roomnumber"]; ?></label>
    </TD>

```

```

        <TD CLASS="formvalue">
        <? echo "<INPUT TYPE=\"text\" NAME=\"roomnumber\" ID=\"froomnumber\"
VALUE=\"$roomnumber\">"; ?> ve
    </TD>
</TR>
<TR>
    <TD CLASS="formlabel"><label accesskey="m" for="fmail">
        <? echo $ldap_attr_str["mail"]; ?></label>
    </TD>
    <TD CLASS="formvalue">
        <? echo "<INPUT TYPE=\"text\" NAME=\"mail\" ID=\"fmail\"
VALUE=\"$mail\">"; ?>
    </TD>
</TR>
<TR>
    <TD CLASS="formlabel">&nbsp;</TD>
    <TD CLASS="formvalue"><INPUT TYPE="submit" NAME="Ara" VALUE="Ara"></TD>
</TR>
</TABLE>
</FIELDSET>
</FORM>

<?
//-----
if (!isset($query)) {
    CommonFooter($footer_links);
    exit;
}
//-----

```


?>

<DIV CLASS="title">Arama Sonuçları</DIV>

<?

//-----

```
$my_con=@ldap_connect(_LDAP_HOST,_LDAP_PORT); // ! this must be a valid
LDAP server !
```

```
if (!$my_con) {
    echo "Hata. LDAP sunucu baglantısı saglanamadı.<BR>Sorun yetkililere
iletildi...";
    mail(getenv("SERVER_ADMIN"), // Target address
        "Ulakbim Telefon Rehberi'nde hata!!!", // Subject
        "LDAP sunucusuna erisim basarisiz. Script: "._FILE_."\nErisim
parametreleri:\nSunucu: "._LDAP_HOST."\nPort:" ._LDAP_PORT."\nSunucu
calismiyor olabilir ya da erisim parametrelerinde hata var...", // Message
body
        "From: ".getenv("SERVER_ADMIN")."\nX-Mailer: PHP/" . phpversion());
    CommonFooter($footer_links);
    exit;
}
```

```
@ldap_bind($my_con); // this is an "anonymous" bind, typically read-
only access
```

```
$base_dn = _ORG;
```

```
if (empty($cn) && empty($telephonenumber) && empty($roomnumber) &&
empty($mail)) {
```

```

$filter = "objectclass=*inetorgperson*";
}
else {
    $filter = "(&(objectclass=*inetorgperson*))";
    if (!empty($cn))          $filter .= "(cn=*.$cn.*)";
    if (!empty($telephonenumber)) $filter .=
"(telephonenumber=*.$telephonenumber.*)";
    if (!empty($roomnumber))  $filter .= "(roomnumber=*.$roomnumber.*)";
    if (!empty($mail))        $filter .= "(mail=*.$mail.*)";
    $filter .= ")";
}

$search_res = ldap_search($my_con, $base_dn, $filter);
$info = ldap_get_entries($my_con, $search_res);

if ($info["count"] == 0) {
    echo "Aranan kriterlere uygun bir kayıt bulunamadi...";
    CommonFooter($footer_links);
    exit;
}

echo "<TABLE WIDTH=\"".$_WIDTH.\"\" BORDER=\"1\" CELLSPACING=\"0\"
CELLPADDING=\"1\">\n";
echo "<TR>\n";

echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["cn"]."</TH>\n";
echo "<TH ALIGN=\"left\"
NOWRAP>".$ldap_attr_str["telephonenumber"]."</TH>\n";
echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["roomnumber"]."</TH>\n";

```

```

echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["ou"]."</TH>\n";
echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["mail"]."</TH>\n";

if (ValidateHOST()) {
    echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["mobile"]."</TH>\n";
    echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["homephone"]."</TH>\n";
    echo "<TH ALIGN=\"left\"
NOWRAP>".$ldap_attr_str["homepostaladdress"]."</TH>\n";
    echo "<TH ALIGN=\"left\" NOWRAP>".$ldap_attr_str["labeleduri"]."</TH>\n";
    echo "<TH ALIGN=\"left\" NOWRAP>&nbsp;</TH>\n";
}

echo "</TR>";

for ($i=0; $i<$info["count"]; $i++) {          // Siralama hadisesi
    $sorted[(string)$i]=@$info[$i]["cn"][0];
}

asort($sorted);

reset($sorted);
while(list($idx, $val)=each($sorted)) {
    echo "<TR>\n";
    echo " <TD NOWRAP>".GetAttr(@$info[$idx]["cn"])."</TD>\n";
    echo " <TD NOWRAP>".GetAttr(@$info[$idx]["telephonenumber"])."</TD>\n";
    echo " <TD NOWRAP>".GetAttr(@$info[$idx]["roomnumber"])."</TD>\n";
    echo " <TD NOWRAP>".FormatLinks(GetAttr(@$info[$idx]["ou"]))."</TD>\n";
    echo " <TD NOWRAP>".FormatLinks(GetAttr(@$info[$idx]["mail"]))."</TD>\n";
}

```

```

if (ValidateHOST()) {
    echo "<TD NOWRAP>".GetAttr(@$info[$idx] ["mobile"])."</TD>\n";
    echo "<TD NOWRAP>".GetAttr(@$info[$idx] ["homephone"])."</TD>\n";
    echo "<TD NOWRAP>".GetAttr(@$info[$idx] ["homepostaladdress"])."</TD>\n";
    echo "<TD
NOWRAP>".FormatLinks(GetAttr(@$info[$idx] ["labeleduri"]))."</TD>\n";
    echo "<TD ALIGN=\"center\" NOWRAP><A
HREF=\"login.php?user=\".GetAttr(@$info[$idx] ["uid"]).\"\"><IMG
SRC=\"edit_pen.gif\" WIDTH=\"15\" HEIGHT=\"10\" BORDER=\"0\"
ALT=\"\".GetAttr(@$info[$idx] ["cn"]).\"\"></A></TD>\n";
    }

    echo "</TR>\n";
}

echo "</TABLE>\n";

ldap_close($my_con);
CommonFooter($footer_links);
?>

```

modify.php - Değişiklik formu üreten scripti

```

<?
/-- Telefon Rehberi --
// modify.php
// LDAP üzerinde degisiklik icin form uretir.
// Yazan: Mustafa Hadi Dilek, Subat 2000
//-----
require("common.php");

```

```
if (!ValidateHOST()) {
    CommonHeader("LDAP Kaydı Değiştirme");
    echo "Üzgünüz. LDAP bilgileri ancak Ulakbim içindeki bilgisayarlardan
değiştirilebilir...";
    CommonFooter($footer_links);
    exit;
}

if (empty($user) || empty($pwd)) {
    if (empty($user))
        @header("Location: login.php?empty=YES");
    else
        @header("Location: login.php?user=".$user."&empty=YES");
    exit;
}

$footer_links["rehber.php"] = "Ulakbim Telefon Rehberi";
$footer_links["login.php"] = "Baska Bir Kullaniciya Gec";

$my_con = @ldap_connect(_LDAP_HOST,_LDAP_PORT);

if (!$my_con) {
    CommonHeader("LDAP Kaydı Değiştirme");
    echo "Hata. LDAP sunucu bağlantısı sağlanamadı.<BR>Sorun yetkililere
iletildi...";
    mail(getenv("SERVER_ADMIN"), // Target address
        "Ulakbim Telefon Rehberi'nde hata!!!", // Subject
        "LDAP sunucusuna erişim başarısız. Script: ".$_FILE_."\nErisim
```

```
parametreleri:\nSunucu: "._LDAP_HOST."\nPort:" ._LDAP_PORT."\nSunucu
calismiyor olabilir ya da erisim parametrelerinde hata var...", // Message
body

    "From: ".getenv("SERVER_ADMIN")."\nX-Mailer: PHP/" . phpversion());
    CommonFooter($footer_links);
    exit;
}

$dn = "uid=".$user.",._ORG;

if (!$r = @ldap_bind($my_con, $dn, $pwd)) {
    @header("Location: login.php?unauth=YES");
    exit;
}

setcookie("user", $user);
setcookie("pwd" , $pwd);

$base_dn    = _ORG;
$filter     = "uid=".$user;
$search_res = ldap_search($my_con, $base_dn, $filter);
$entries    = ldap_get_entries($my_con, $search_res);
$info       = $entries[0];

CommonHeader("LDAP Kaydı Değiştirme");
//-----
?>
<FORM ACTION="modify_res.php" METHOD="POST" ENCTYPE="application/x-www-form-
urlencoded">
```

```

<?
//-----
echo "<TABLE WIDTH=\"".$_WIDTH.\"\" BORDER=\"0\" CELSPACING=\"1\"
CELLPADDING=\"1\">\n";

reset($ldap_attr_str);
while(list($key, $val)=each($ldap_attr_str)) {
    if (!in_array($key, array_keys($info))) {
        echo "Kullanıcının ".$val." özelliği yok...<BR>\n";
        continue;
    }
//-----
?>
<TR>
    <TD CLASS="formlabel"><? echo $val; ?>:&nbsp;</TD>
    <TD CLASS="formvalue">
<?
//-----
    if (in_array($key,$ldap_attr_changeable)) {
        echo "<INPUT NAME=\"".$key.\"\" TYPE=\"Text\"
VALUE=\"".$_GetAttr(@$info[$key]).\"\"></INPUT>";
    }
    else {
        echo GetAttr(FormatLinks(@$info[$key]));
    }
//-----
?>
</TD>
</TR>

```

```

<?
//-----
    } //while
//-----
?>

<TR>

    <TD CLASS="formlabel">&nbsp;</TD>

    <TD CLASS="formvalue">

        <INPUT NAME="submit_change" TYPE="Submit" VALUE="Değiştir"></INPUT>

        <INPUT NAME="submit_cancel" TYPE="Submit" VALUE="İptal"></INPUT>

    </TD>

</TR>

</TABLE>

</FORM>

<?
//-----

    ldap_close($my_con);

    $footer_links["login.php"] ="Başka Bir Kullanıcıya Gec";
    CommonFooter($footer_links);
?>

```

modify_res.php - Değişikliği uygulayıp sonuçları gösteren script

```

<?
//-- Telefon Rehberi --

// modify_res.php

// LDAP uzerinde degisiklik yapip sonuclari kullaniciya bildirir.

// Yazan: Mustafa Hadi Dilek, Subat 2000

//-----

```



```
require("common.php");

if (!ValidateHOST()) {
    CommonHeader("LDAP Değişiklik Sonuçları");
    echo "Üzgünüz. LDAP bilgileri ancak Ulakbim içindeki bilgisayarlardan
değiştirilebilir...";
    CommonFooter($footer_links);
    exit;
}

$footer_links["rehber.php"] = "Ulakbim Telefon Rehberi";
$footer_links["modify.php"] = "LDAP Kaydı Değiştirme";
$footer_links["login.php"] = "Başka Bir Kullanıcıya Geç";

if (!isset($submit_change) && !isset($submit_cancel)) {
    @header("Location: login.php?empty=YES");
    exit;
}

if (!isset($HTTP_COOKIE_VARS["user"]) || !isset($HTTP_COOKIE_VARS["pwd"])) {
    @header("Location: login.php?enablecookie=YES");
    exit;
}

$user = $HTTP_COOKIE_VARS["user"];
$pwd = $HTTP_COOKIE_VARS["pwd"];

if ($user=="" || $pwd=="") {
    @header("Location: login.php?empty=YES");
```

```
    exit;
}

if(!empty($submit_cancel)) {
    echo "Değişiklik iptal edildi...\n";
    CommonFooter($footer_links);
    exit;
}

elseif(!empty($submit_change)) {
    $my_con = @ldap_connect(_LDAP_HOST,_LDAP_PORT);

    if (!$my_con) {
        CommonHeader("LDAP Degisiklik Sonuclari");
        echo "Hata. LDAP sunucu baglantısı saglanamadı.<BR>Sorun yetkililere
iletildi...";

        mail(getenv("SERVER_ADMIN"), // Target address
            "Ulakbim Telefon Rehberi'nde hata!!!", // Subject
            "LDAP sunucusuna erisim basarisiz. Script: \"._FILE_\".\nErisim
parametreleri:\nSunucu: \"._LDAP_HOST.\".\nPort: \" . _LDAP_PORT.\".\nSunucu
calismiyor olabilir ya da erisim parametrelerinde hata var...\", // Message
body
            "From: \" .getenv("SERVER_ADMIN").\".\nX-Mailer: PHP/" . phpversion());

        CommonFooter($footer_links);
        exit;
    }

    $dn = "uid=".$user.", "_ORG;
```

```

if(!($r = @ldap_bind($my_con, $dn, $pwd)) {
    @header("Location: login.php?unauth=YES");
    exit;
}

$base_dn    = $dn;
$filter     = "objectClass=*";           // tum entryleri
getirir...

$search_res = ldap_read($my_con, $base_dn, $filter);
$info       =
ldap_get_attributes($my_con, ldap_first_entry($my_con, $search_res));

reset($ldap_attr_str);

while(list($key, $val)=each($ldap_attr_str)) {
    if (in_array($key, $ldap_attr_changeable)) {
        if (in_array($key, array_keys($info))) {
            $$key = trim($$key);

            if (empty($$key)) $$key="-";

            if (is_array(@$info[$key])) {
                if (@$info[$key][0] != $$key) $changes[$key] = $$key;
            }
            else {
                if (@$info[$key] != $$key) $changes[$key] = $$key;
            }
        }
    }
}
}

```

```
}

CommonHeader("LDAP Değişiklik Sonuçları");

if (is_array(@$changes)) {
    if (in_array("givenname",array_keys(@$changes)) ||
in_array("sn",array_keys(@$changes))) {
        if (in_array("givenname",array_keys(@$changes)))
            $changes["cn"] = $changes["givenname"];
        else
            $changes["cn"] = @$info["givenname"][0];

        $changes["cn"] .= " ";

        if (in_array("sn",array_keys(@$changes)))
            $changes["cn"] .= $changes["sn"];
        else
            $changes["cn"] .= @$info["sn"][0];
    }

    my_ldap_modify($dn, $pwd, $dn, $changes);

    reset($changes);
    while(list($key,$val)=each($changes))
        echo $ldap_attr_str[$key]." <B>".$changes[$key]."</B> olarak
degistirildi...<BR>\n";
    }
else
    echo "Kaydınızda değişiklik yapılmadı...<BR>\n";
}
```

```
        ldap_close($my_con);
    } // elseif

    CommonFooter($footer_links);
?>
```

common.php - Ortak fonksiyon kütüphanesi

```
<?
//-- Telefon Rehberi --
// common.php
// Ortak olarak kullanılacak fonksiyonlar ve konfigürasyon opsiyonları içerir.
// Bu dosya webden okunamamalı veya çağrılmamalı
// Yazan: Mustafa Hadi Dilek, Subat 2000

include("conf.php"); // Konfigürasyon dosyası

define("_CRLF", "\\n");
define("_QUOTE", "\"");

//-----
//--- Herhangi LDAP client yazılımında bulunan ldapmodify programını
kullanarak
//--- LDAP entriesinde değişiklik yapar
//--- Input parametreleri:
//----- Değişikliği yapmaya yetkili kullanıcının (dn) bilgisi (admin veya
kullanıcının kendisi olabilir),
//----- Değişikliği yapmaya yetkili kullanıcının LDAP sunucusundaki şifresi,
//----- Üzerinde değişiklik yapılacak kullanıcının distinguished name (dn)
```

```

bilgisi,
//----- Degisiklikleri iceren bir hash array
//--- Burada bir modify stringi, standart ldif formatinda olusturulur.
//--- Bu format 'man ldapmodify' ile gorulebilir.
//--- Veya Michigan Universitesinin ldap manual web sayfasinda ayrintili
olarak var.
//--- www.umich.edu dan aranabilir.

function my_ldap_modify($auth_dn, $auth_pwd, $dn, $changes) {

    $modifystr = _QUOTE.
        "dn: ".$dn._CRLF.
        "changetype: modify"._CRLF;

    while(list($key,$val)=each($changes)) {
        $modifystr = $modifystr.
            "replace: ".$key._CRLF.
            $key.": ".$val._CRLF.
            "-"._CRLF;
    }

    $modifystr = $modifystr._QUOTE;

    exec("printf ".$modifystr." | "._LDAP_DIR._LDAP_MODIFY." -h "._LDAP_HOST."
-p "._LDAP_PORT." -D ".$auth_dn." -w ".$auth_pwd);
}
//-----
//--- Sayfalarin ust tarafinda yer alacak kismi olusturur:

function CommonHeader($Title) {

```

```
?>
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-8859-9">
  <LINK HREF="http://yeni.ulakbim.gov.tr/css/extranetGenel.css"
REL="stylesheet" TYPE="text/css">
  <? include("rehber.css"); ?> <!-- Stil dosyasi -->
  <TITLE><? echo $Title ?></TITLE>
</HEAD>

<BODY BGCOLOR="White">
  <DIV CLASS="title"><? echo $Title ?></DIV>
<?
  }
//-----
//--- Sayfalarin alt tarafinda yer alacak kismi olusturur:
//--- Input parametresi anahtari html linki; degeri ise linke ait metin olan
associative arraydir
function CommonFooter($Footer) {
  if (!empty($Footer) && is_array($Footer)) {
    echo "\n<DIV CLASS=\"footer\">";
    reset($Footer);
    while(list($Link, $Text)=each($Footer)) {
      echo "\n<A HREF=\"\$Link\">[$Text]</A>&nbsp;";
    }
    echo "\n</DIV>";
  }

  echo "\n</BODY>\n";
}
```

```

    echo "</HTML>\n";
}
//-----
// URL ve mail adreslerini link olarak basar. Dikkatli kullanilmali
function FormatLinks($str) {
    $str = preg_replace("/(\w+):\/\//([^\ ]+)/", "<A
HREF=\"\\1:\\2\">\\1:\\2</A>", $str); // her tur URL icin
    $str = preg_replace("/(\w+)@([^\ ]+)/", "<A
HREF=\"mailto:\\1@\\2\">\\1@\\2</A>", $str); // mail icin
    return $str;
}
//-----
// Verilen LDAP attribute'unu basilabilecek halde alir:
function GetAttr($attr) {
    if (is_array($attr)) $outstr = trim(@$attr[0]);
    else $outstr = trim($attr);

    return ($outstr != "") ? $outstr : "&nbsp;";
}
//-----
// Remote makine internal networkumuzun icinde ise 'True' dondurur.
function ValidateHOST() {
    if (preg_match("/^"._INTERNAL_NET."\.[0-9]+$/", getenv("REMOTE_ADDR")))
        return True;
    else
        return True;
}
?>

```


PHP ile Mail, Imap, POP3, NNTP

Uygulamaları

SMTP, IMAP, POP3, NNTP birbirleriyle yakından ilişkili protokollerdir.

SMTP:

SMTP(Simple Mail Transfer Protocol) mail gönderme ve almada kullanılan protokoldür ve UNIX ortamında bu iş için en yaygın kullanılan araç sendmail'dir. PHP, UNIX ortamında kullanılacak ise mail almak için aynı makinada bir sendmail sunucusunun çalışır durumda olması gerekmektedir. Mail gönderirken sendmail otomatik olarak çalıştırıldığı için sunucunun çalışır durumda olmasına gerek yoktur.

- mail fonksiyonu mail göndermek için kullanılır. Bu fonksiyonda dördüncü parametre ek SMTP header bilgisi göndermek için kullanılır. Eğer kullanılmaz ise mesajın göndereni kısmına web sunucusunun uid'i otomatik olarak konulur (Örnek: nobody@abc.def.com).

Ek header bilgileri şunlar olabilir:

- From: <Gönderilen adres>
- Reply-To: <Gönderen adres>
- X-Mailer: <Maili gönderen istemci (client) program>

IMAP:

IMAP(Interet Message Access Protocol), SMTP ile mail sunucusna alınan maillere kullanıcıların uzaktan erişimini sağlayan bir protokoldür. Gelen postaların sunucu tarafında merkezi olarak tutulması, kullanıcının burada birden çok dizin açmasına izin vermesi, mesajları çok parçalı olarak işleyebilme, eşzamanlı erişim ve dizin paylaşımı gibi özellikleri ile POP3'e göre daha üstün bir protokoldür. PHP ile IMAP kullanımı kabaca şu adımları içerir:

- **Posta sunucu ile bağlantı kurulur:**

- `imap_open` fonksiyonu, verilen posta kutusu tanımlayıcısına sahibinin (kullanıcı adı ve şifresi ile) bağlanabilmesini sağlar. Bağlantı sağlanırsa posta kutusunu ifade eden bir numara (handle) döndürülür. Burada mailbox, "`{<Sunucu makina adresi>/imap:<imap portu><posta kutusu dizini>`" şeklinde bir stringdir. Örnek: `{javaserver.ulakbim.gov.tr/imap:143}INBOX`
- `imap_reopen` fonksiyonu, handle'ı verilen posta kutusu kapatılmış ise aynı HTTP oturumu içinde doğrulama yapılmaksızın yeniden bağlanılabildiğini sağlar.

- **Mesaj başlık (header) ve içerik bilgisine ulaşılabilir:** IMAP, mesajları mesaj sıra numarası (sequence) ve tanımlayıcı numara (uid) olmak üzere iki tür numaralama ile ayırır. Her ikisi de PHP ile mesaj bilgilerine ulaşmak için kullanılabilir. Ancak sequence kullanmak birbirinden ayrı oturumlar kullanan HTTP için çok sağlıklı değildir. Çünkü posta kutusundan bir mesaj silindiği anda sequence numaraları yeniden düzenlenir. Ancak uid değerleri değişmez.

- `imap_headers` fonksiyonuyla, verilen posta kutusundaki mesaj başlıkları sıra

numarasına göre (sequence) sıralı bir dizi olarak alınır.

- `imap_header` fonksiyonu, posta kutusundaki verilen sıra numaralı mesajın ayrıntılı çözümlenmiş (parsed) header bilgisi object olarak alınır.
- `imap_body` fonksiyonu, posta kutusundaki verilen sıra numaralı mesaj içeriği alınır. Burada `FT_UID` flagı olarak üçüncü bir parametre ile çağrılırsa sıra numarası (sequence) yerine tanımlayıcı (uid) kullanır.
- **Arama ve sıralama işlemleri yapılabilir:**
 - `imap_sort` fonksiyonu, posta kutusundaki mesajların verilen kritere göre sıralanmış listesi dizi olarak alınır. Burada `options` kısmında `SE_UID` kullanılırsa dizi elemanları uid değerlerinden oluşur aksi takdirde sequence numaraları döndürülür
 - `imap_search` fonksiyonu, posta kutusundaki mesajlar üzerinde verilen kritere göre arama yapılır. Burada da `flags` kısmında `SE_UID` kullanılırsa sequence numaraları yerine uid değerleri alınır.

POP3

POP3(Post Office Protocol) protokolü de IMAP gibi, SMTP ile mail sunucusuna alınan maillere kullanıcıların uzaktan erişimini sağlayan bir protokoldür. Ancak mesajların kullanıcılar tarafından bütün olarak download edilmesini gerektirmesi, merkezi olarak tutulmadığı için hareket halindeki kullanıcılar (roaming users) için senkronizasyon sorunları oluşturması ve eşzamanlı erişime izin vermemesi nedeniyle IMAP'ın gerisinde kalır.

PHP ile POP3 kullanımını IMAP'dan çok farklı değildir. Fakat, IMAP'ın izin verdiği bazı

fonksiyonlar POP3 erişiminde kullanılamaz.

Bağlantı kurarken ilgili protokol adını ve kullanılan portu örneğin şu şekilde değiştirmek yeterli olur: {javaserver.ulakbim.gov.tr/pop3:110}INBOX

NNTP:

NNTP (Network News Transfer Protocol) protokolü, İnternet üzerinde haber mesajlarının merkezi olarak sunucular üzerinde tutulup tutulup, istemcilerin bu kaynaklar üzerinde sorgulama yapması, mesajları okuması ve mesaj atabilmesini sağlayan bir protokoldür.

Bağlantı kurarken ilgili protokol adını ve kullanılan portu örneğin şu şekilde değiştirmek yeterli olur: {javaserver.ulakbim.gov.tr/nntp:119}INBOX

PHP ile havada grafik üretme

GD Image Library

PHP grafik üretmek için bu kütüphaneye gereksinim duyuyor. Adres:

<http://www.boutell.com/gd/>

Yapılabilecekler:

İmaj yaratma:

PHP ile dosyadan varolan bir imaj alınabilir veya yeni bir imaj yaratılabilir. PHP her iki durumda da hafızada bir imaj nesnesi üreterek bunu bir handle ile bize bildirir:

- *int imagecreate(int x_size, int y_size)*: Verilen ölçülerde yeni bir imaj nesnesi yaratarak bu imaja ait handle'ı döndürür.
- *int imagecreatefrompng(string filename)*, *int ImageCreateFromJpeg(string filename)*: PNG veya Jpeg formatındaki dosyayı okuyarak yeni yarattığı bir imaj nesnesi içine alır ve yaratılan imajın handle'ını döndürür.

İmaj bilgilerini alma:

getimagesize fonksiyonu imaja ilişkin bazı bilgiler içeren bir dizi döndürür. Bu dizinin elemanları:

- Index 0: İmajın genişliği
- Index 1: İmajın yüksekliği
- Index 2: İmajın türü (1:gif 2:jpeg 3:png)
- Index 3: "height=xxx width=xxx" şeklinde doğrudan HTML kodu içinde kullanılacak string

İmaj üzerinde değişiklik yapma:

- Renk belirlemeleri:
 - *imagecolorallocate* fonksiyonu ile renk atamaları yapılır.
 - *imagecolortransparent* fonksiyonu ile transparent renk belirlemesi yapılabilir. Gif ve PNG formatları transparent özelliğini desteklemektedir.
- İmaj üzerinde şekil üretme:

- imageline
- imagearc
- imagepolygon
- imagepolygon
- imagerectangle
- imagefill
- İmaj üzerinde metin üretme:
 - imagestring
 - imagettftext
 - imagettfbbox

İmaj çıktısı üretme:

Buraya kadar tüm önbelirlemeleri yapılan imaj nesnemiz bu aşamada uygun algoritma ile istenen formata çevrilerek browsere gönderilir. Buradaki fonksiyonlarda dosya adı girilirse çıktı browsere gönderilmeyip dosyaya yazılır.

- imagegif fonksiyonu, imaj numarası (handle) ile verilen imajı GIF formatında üreterek browsere gönderir.
- imagejpeg fonksiyonu, imaj numarası (handle) ile verilen imajı JPEG formatında üreterek browsere gönderir

- imagepng fonksiyonu, imaj numarası (handle) ile verilen imajı PNG formatında üretmek için kullanılır ve oluşturulan PNG formatındaki imajı browser'e gönderir.