

Morgado-Samagaio Jonathan

2A

PWS / TP1 PHP

Table des matières

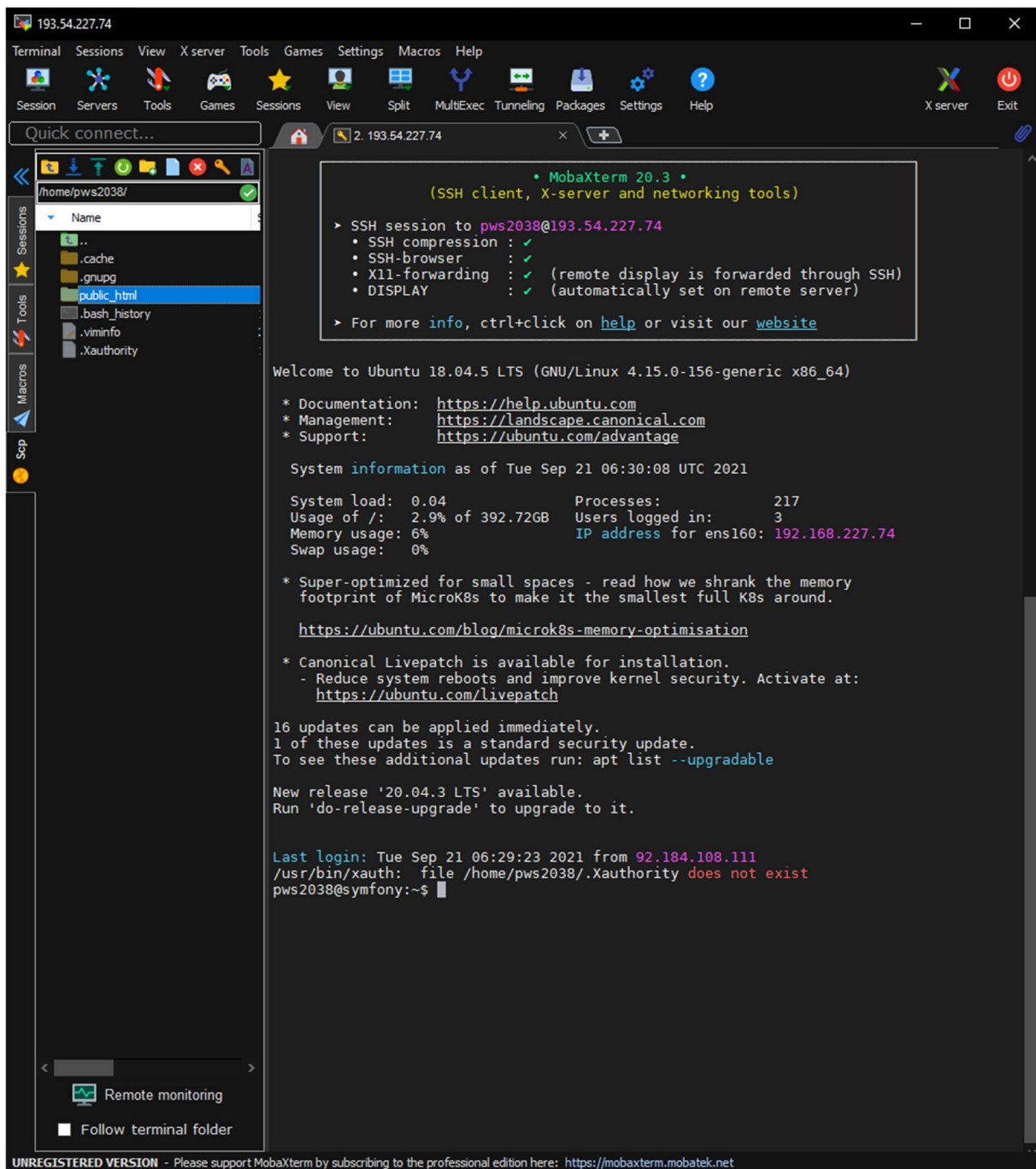
Table des matières	2
1. Introduction.....	3
2. MobaXtern.....	3
3. PHP	4
4. Conclusion	9

1. Introduction

Dans ce TP, nous allons voir comment utiliser MobaXterm pour se connecter au serveur et aussi les bases du PHP.

2. MobaXterm

Nous commençons par nous connecter avec nos identifiants de base et nous changeons notre mot de passe. Nous arrivons sur cette page :



```
193.54.227.74
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 193.54.227.74
/home/pws2038/
Name
.cachef
.gnupg
public_html
.bash_history
.viminfo
.Xauthority

MobaXterm 20.3
(SSH client, X-server and networking tools)
> SSH session to pws2038@193.54.227.74
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✓ (remote display is forwarded through SSH)
  • DISPLAY : ✓ (automatically set on remote server)
> For more info, ctrl+click on help or visit our website

Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-156-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Tue Sep 21 06:30:08 UTC 2021

System load: 0.04 Processes: 217
Usage of /: 2.9% of 392.72GB Users logged in: 3
Memory usage: 6% IP address for ens160: 192.168.227.74
Swap usage: 0%

* Super-optimized for small spaces - read how we shrank the memory footprint of MicroK8s to make it the smallest full K8s around.
https://ubuntu.com/blog/microk8s-memory-optimisation

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

16 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Sep 21 06:29:23 2021 from 92.184.108.111
/usr/bin/xaauth: file /home/pws2038/.Xauthority does not exist
pws2038@symfony:~$
```

Nous créons ensuite l'arborescence attendu dans public_html.

3. PHP

En prenant l'archive sur Moodle et en ouvrant les fichiers à l'intérieur nous tombons sur cette page :



Nous allons changer quelques éléments de la page en modifiant les fichiers footer.php, header.php et menu.php :

```
//Modification de "page 2" Par AffecterCategories
echo '<li class="nav-item">';
if ($page == 'AffecterCategories.php') {
    echo '<a class="nav-link active" href="AffecterCategories.php">AffecterCategories</a>';
}
else {
    echo '<a class="nav-link" href="AffecterCategories.php">AffecterCategories</a>';
}
echo '</li>';
```

Et nous obtenons cette page :



Nous allons maintenant écrire le contenu de AffecterCategories.php. Voici le code :

```
<div style="padding-top: 30px" id="main">
  <div style="text-align: center" class="col-md-12">
    <?php
      $sexe = "H";
      $age = 19;
      $poids = 80;
      $genre = ($sexe == "F") ? "Mme" : "Mr";
      $catAge = ($age < 21) ? "junior" : "senior";

      switch (true)
      {
        case $poids < 60:
          $catPoids = "Super-léger";
          break;
        case $poids < 73:
          $catPoids = "Légers";
          break;
        case $poids < 90:
          $catPoids = "\"super\" moyens";
          break;
        case $poids < 100:
          $catPoids = "mi-lourds";
          break;
        case $poids >= 100:
          $catPoids = "lourds";
          break;
      }

      echo "$genre est un $catAge dans la catégorie $catPoids";
    ?>
  </div>
</div>
```

Nous pouvons voir les bases de la syntaxe du php avec le \$ devant le nom des variables. Nous avons aussi le Switch case comme en java et le echo pour écrire sur la page HTML.

Nous écrivons maintenant AfficheTitres.php. Voici le code :

```
<div style="padding-top: 30px" id="main">
  <div style="text-align: center" class="col-md-12">
    <?php
      define ("NBNIIVTITRES", 6);
      for ($i = 0; $i < NBNIIVTITRES; $i++)
      {
        echo "<h$i>$i: Titre de niveau $i</h$i>";
      }
    ?>
  </div>
</div>
</div>
```

Nous pouvons voir le define pour définir des constantes et la formation d'une boucle for.

Voici le rendu :

[Accueil](#)

[AffecterCategories](#)

[AfficheTitres](#)

[Test_multiplie](#)

0: Titre de niveau 0

1: Titre de niveau 1

2: Titre de niveau 2

3: Titre de niveau 3

4: Titre de niveau 4

5: Titre de niveau 5

Il nous reste maintenant à faire Fonctions.php pour faire multiplie(). Voici le code :

```

function multiplie($taille = 5)
{
    for($i = 0; $i < $taille+1; $i++)
    {
        for ($j = 0; $j < $taille+1; $j++)
        {
            $tab[$i][$j] = 0;
            if ($i == 0 && $j == 0) {$tab[$i][$j] = "X";}
            else if ($i == 0) {$tab[$i][$j] = $j;}
            else if ($j == 0) {$tab[$i][$j] = $i;}
            else
            {
                $tab[$i][$j] = $tab[0][$j] * $tab[$i][0];
            }
        }
    }
}

```

Première partie avec la définition de la fonction et son paramètre par défaut. Il y a aussi l'utilisation d'une boucle for imbriquée permettant de parcourir un tableau pour la création de la table de multiplication.

```

echo "<h2> Tableau de multiplication </h2>";
echo "<table border =\'2\'>";

for ($i = 0; $i < $taille+1; $i++)
{
    echo "<TR>";
    for ($j = 0; $j < $taille+1; $j++)
    {
        $temp = $tab[$i][$j];
        if ($j == 0 || $i == 0)
        {
            echo "<TH>$temp</TH>";
        }
        else
        {
            echo "<TD>$temp</TD>";
        }
    }
    echo "</TR>";
}

echo "</table>";

```

La seconde partie avec la création du tableau HTML.

4. Conclusion

Dans ce TP, nous avons pu utiliser les bases du PHP pour créer des pages web dynamiques. Nous avons vu comment utiliser des variables, des conditions, des boucles et des tableaux. Nous avons aussi vu comment utiliser MobaXtern pour envoyer nos fichiers PHP avec le serveur.