

# TP révision

Les DNS (*Domain Name Server*) sont une infrastructure centrale et nécessaire au bon fonctionnement d'internet. C'est grâce à eux que vous pouvez accéder à google.com en cherchant `https://google.com/` plutôt qu'en cherchant `https://142.250.178.142/`

Le DNS doit être capable de :

- Recevoir une requête contenant un nom de domaine (par exemple `adh6.minet.net` , et de renvoyer l'adresse IP correspondant
- Mettre à jour sa base de donnée d'adresse IP avec un nouveau nom de domaine
- Faire une recherche inversée (retrouver le nom de domaine à partir de l'adresse IP)

---

## Gestion des fichiers

Pour des raisons de simplicité, on stockera les adresses IPs dans des dossiers et fichiers dont les noms correspondront au domaine.

### Note

Les noms de domaines se divisent en sous-domaines séparés par des points.

Par exemple le nom de domaine `adh6.minet.net` est en fait le domaine `adh6` qui fait partie du domaine `minet`, lui même faisant partie du domaine `net`

## Les updates

Avant de pouvoir effectuer des *requêtes*, il faudrait tout d'abord remplir notre DNS grace à des *updates*

1. Créez un fichier `nsupdate.sh` et attribuez vous les droits d'execution dessus
2. La syntaxe du programme sera `./nsupdate [adresse ip] [domaine]`, vérifiez au début de votre programme que le bon nombre d'argument est donné ( $\geq 2$ ) et renvoyez un message d'erreur si la syntaxe n'est pas respectée.

Pour simplifier, on va supposer que le nom de domaine est donné dans l'ordre inverse et les sous noms sont séparés par des espaces

(`www.minet.net` deviendra `./nsupdate.sh 157.159.40.103 net minet www`)

3. Créez une variable `ip` contenant l'adresse ip fournie, puis décalez les arguments.

Pour chaque sous-nom de domaine, il faut d'abord vérifier si le dossier correspondant n'existe pas déjà.

4. Itérez sur les arguments restant. À chaque argument : vérifiez si le dossier n'existe pas, le cas échéant créez le puis entrez dans le dossier
5. Une fois dans le bon dossier, écrivez l'adresse ip dans un fichier nommé "address"

## Les requêtes

Maintenant que nous avons rempli notre "base de données", on s'intéresse maintenant à la manière d'y accéder.

1. Créez un fichier `nslookup.sh` et attribuez vous les droits d'execution dessus et vérifiez que la syntaxe `nslookup.sh [domaine]` est respectée
2. En utilisant une méthode similaire à celle de la première partie, écrivez un programme qui affiche sur la sortie standard l'adresse ip correspondant au nom de domaine.

Si le nom de domaine n'est pas reconnu (le fichier n'existe pas), le programme doit afficher un message d'erreur (par exemple `domain not found`) et renvoyer un code d'erreur différent de 0

On veut que plusieurs utilisateurs puissent utiliser le DNS en même temps, il faut donc éviter un conflit entre lecture et écriture (lecture d'une donnée à moitié écrite) ou deux écritures concurrentes.

3. Implémentez des sémaphores (P.sh et V.sh) afin d'empêcher ces problèmes.

## La recherche inversée

La recherche inversée est plus utile qu'elle n'y paraît, en effet certains fournisseurs de messagerie électronique (email) refusent, pour des raisons de sécurité, tout message provenant d'une IP non reconnue par une recherche inversée.

1. Créez un fichier `nslookup.sh` et attribuez vous les droits d'exécution dessus et vérifiez que la syntaxe `nslookup.sh [adresse ip]` est respectée

Pour rechercher l'adresse IP correspondante, on va tout d'abord itérer sur le contenu du répertoire courant et, en fonction du type (fichier ou dossier), on va :

- dans le cas d'un fichier : vérifier si le contenu correspond à l'IP cherchée
  - dans le cas d'un dossier : répéter la même opération récursivement dans ce dossier
2. Implémentez cet algorithme (on imprimera simplement le chemin du fichier où on trouve l'adresse, pensez aux sémaphores)

### Note

On a ici implémenté un algorithme de recherche dans un arbre appelé DFS (Depth-First Search) bien connu des anciens MP option info 😊.

Notez que cet algorithme peut se généraliser (avec une petite modification) au cas d'un graphe

## Bonus

Les DNS permettent aussi des redirections, par exemple `adh6.minet.net` renvoie en réalité sur `www.minet.net` (qui lui même va s'occuper de faire la bonne redirection). Il peut être intéressant d'implémenter ces redirections avec des liens (avec la commande `ln`)