

cExercices 2023-2024 Provisoire v0,9
Compléments Réseaux
2 Informatique Système Option Telecom Quatrième quart

Introduction

Prérequis

Dans le dossier complement Reseaux
Se munir du tome DNS 2018 Traduct.

Plateforme de développement

Toute variante de Unix acceptant les librairies du troisième quart (**Lib2022DnsEtud**). Je ne peux cependant garantir le fonctionnement que sur SUN & ubuntu. En aucune manière, le fait que votre plateforme de développement soit en panne le jour de l'examen, ne peut constituer une excuse pour ne pas présenter l'examen. Il vous restera toujours la solution ubuntu.

Objectif du laboratoire

Ecrire un serveur DNS *bind 8 et* un client *nslookup* élémentaire

Personnalisation du programme

- Tout les noms de programme doivent se terminer pas vos initiales si vous travaillez seul ou les deux noms des coéquipier (Exemple, BorneEHV).
- Les structures de données doivent se terminer par les initiales.
- Même principe pour toutes les fonctions ou procédures définies.
- Chaque programme quand il se lance doit afficher le nom et prénom de l'étudiant

Planning des travaux

Dossier à remettre par mail semaine du , Dimanche minuit au Plus tard	Plannning	Echéance
25 / 3		
15 / 4		1,2,3
24 / 4		
22/ 4		4,4B,5
8 / 5		
13/5		6,7,8
Examen	,	Présenter 9,10,11,12,13
Examen		M'envoyer le source de dnslib.c avant L'examen

A l'examen Présenter oralement , Exercices 10 à 14

1) Parser et lire un fichier ligne par ligne

Expliquez ce que font les deux programmes suivants

ex01.c

```
#include <stdio.h>
#include <string.h>

main()
{
    char B1[30],B2[30],B3[30],BB[30],BB2[30] ;
    int d1,d2,d3 ;

    char *Buffer = "primary 249.249.192.in-addr.arpa db.192.249.249" ;
    char *Buffer2="249.192.in-addr.arpa" ;

    bzero(B1,sizeof B1 ) ; bzero(B1,sizeof B2 ) ; bzero(B1,sizeof B3 ) ;
    sscanf(Buffer,"%30s %30s ",B1,B2) ;
    printf("T1 %s %s \n",B1,B2) ;
    bzero(B1,sizeof B1 ) ; bzero(B1,sizeof B2 ) ; bzero(B1,sizeof B3 ) ;
    sscanf(Buffer,"%30s %30s %30s",B1,B2,B3) ;
    printf("T2 %s %s %s \n",B1,B2,B3) ;
    sscanf(B2,"%d.%d.%d.%30s",&d1,&d2,&d3,BB) ;
    printf(" %d %d %d %s\n",d1,d2,d3,BB) ;

    d1 = d2 = d3 = 0 ;
    sscanf(Buffer2,"%d.%d.%d.%30s",&d1,&d2,&d3,B3) ;
    printf(" %d %d %d %s\n",d1,d2,d3,BB) ;

    sscanf(Buffer2,"%d.%d.%30s",&d1,&d2,B3) ;
    printf(" %d %d %s\n",d1,d2,BB) ;
}
```

ex02.c

```
#include "stdio.h"

main()
{
    FILE *fp ;
    char Buffer[81] ;

    fp = fopen("ex01.c","r") ;
    while ( fgets(Buffer,sizeof Buffer,fp),!feof(fp) )
    {
        printf("%s",Buffer) ;
    }
    fclose(fp) ;
}
```

2 Rechercher l'ip pour un nom donné

En utilisant fgets et sscanf de l'exemple précédent,

On demande d'entre un nom et de rechercher l'ip correspondante dans le fichier hosts suivant :

```
127.0.0.1    localhost

# Ceci sont nos machines les plus performantes
# -----

192.249.249.2 rocobop.film.com      robocop
192.249.249.3 terminator.film.com  terminator  bigt
192.249.249.4 diehard.film.com     diehard     dh

#
# Les machines les moins performantes du reseau
#

192.253.253.2 misery.film.com      misery
192.253.253.3 shining.film.com     shining
192.253.253.4 carrieL.film.com      carrie

#
# La configuration du routeur
#

192.249.249.1 gateway.film.com      gateway     gw     gw249
192.253.253.1 gateway.film.com      gateway     gw     gw253
```

Attention dans le sscanf à bien faire correspondre la dimension avec la taille de vos buffer sous peine d'avoir de mauvaises surprises

```
vanstapel@vanstap2:~/lib2022G/D2$ ./ex02
>terminator.film.com
Search:terminator.film.com:longueur 19
Le resultat 192.249.249.3:
```

3) Choisir un nom de domaine et un de sous-domaine personnel (*)

Ecrire les fichiers suivant pour votre domaine et sous-domaine
named.conf,
nom vers ip
ip vers nom. (inutile si sous-domaine)

Ecrire les fichiers suivant pour votre sous-domaine
named.conf,
nom vers ip

Il n'y a pas de code dans cette étape.

Exemple :

Nom de domaine. Famille.dupont.eu
Nom de sous-domaine Femmes.Famille.Dupont.eu

(*) Celui donné dans le cours ne peut pas être repris !!!!!

4) Télécharger le rar Lib2022GDNSStu.tar dans le dossier Complement Reseaux\ labo. Compiler et exécuter l'exemple dans **ListeS**.

Comprendre et savoir expliquer l'usage de la liste linéaire `sll_s` cette structure de donnée sera utilisée dans les étapes suivantes. J'attire votre attention sur `liste_print`.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../ndnslib/dnslib.h"

struct Data {
    int Compteur ;
    char Message[80] ;
} ;

void list_print (sll_s * p_l)
{
    int i;
    int size ;

    //sll_save(p_l) ;
    size = sll_sizeof (p_l);
    fprintf(stderr,"Taille %d \n", size ) ;

    sll_first (p_l);
    for (i = 0; i < size; i++)
    {
        if (sll_data (p_l) != NULL)
        {
            struct Data *pData ;
            pData = ( struct Data *) sll_data (p_l) ;
            printf ("%d %s\n", pData->Compteur , pData->Message );
        }
        sll_next (p_l);
    }
    //sll_restore(p_l) ;
    printf ("\n");
}

int main (void)
{
    sll_s *p_sll = NULL;
    struct Data *pData ;
    int Compteur ;

    Compteur = 0 ;
    printf ("\t-- Initialisation --\n");
    p_sll = sll_new ();
    list_print (p_sll);
    printf ("\n\t-- Insertion --\n");
    {
        int i;
        for (i = 0; i < 4; i++)
        {
            int taille ;
            pData = (struct Data * ) malloc( sizeof(struct Data) ) ;
            pData->Compteur=i ;
            strcpy(pData->Message,"coucou") ;
            sll_insert (p_sll, pData ) ;
        }
    }
```

```

taille = sll_sizeof (p_sll) ;
printf("La taille est %d\n",taille );
//sll_first (p_sll);
}
}
list_print (p_sll);
printf ("\n\t-- Suppression -du premier élément-\n") ;
sll_first (p_sll) ;
sll_removeFirst (p_sll);
list_print (p_sll);

printf ("\n\t-- Ajout d'un élément après le premier \n") ;
pData = (struct Data * ) malloc( sizeof(struct Data) ) ;
pData->Compteur=5 ;
strcpy(pData->Message,"coucou 5") ;
sll_first (p_sll) ;
sll_insert (p_sll, pData ) ;
list_print (p_sll);

printf ("\n\t-- Ajout d'un élément après le dernier \n") ;
pData = (struct Data * ) malloc( sizeof(struct Data) ) ;
pData->Compteur=6 ;
strcpy(pData->Message,"coucou 6") ;
sll_last (p_sll) ;
sll_insert (p_sll, pData ) ;
list_print (p_sll);

printf ("\n\t-- Destruction --\n");
sll_delete (&p_sll);

return EXIT_SUCCESS;
}

```

Le résultat à l'écran donne

```

anstapel@vanstap2:~/lib2022DnsEtud/ListeSS$ ./main
-- Initialisation --
Taille 0

-- Insertion --
La taille est 1
La taille est 2
La taille est 3
La taille est 4
Taille 4
0 coucou
1 coucou
2 coucou
3 coucou

-- Suppression -du premier élément-
Taille 3
1 coucou
2 coucou
3 coucou

```

	-- Ajout d'un élément après le premier	
Taille 4		
1 coucou		
5 coucou 5		
2 coucou		
3 coucou		
	-- Ajout d'un élément après le dernier	
Taille 5		
1 coucou		
5 coucou 5		
2 coucou		
3 coucou		
6 coucou 6		
	-- Destruction --	

4B)

Copier les éléments de ListeS dans D4

Dans l'exemple précédent, remplacer la structure data par la suivante

struct Zone

```
{
    char nom[80] ;      // Nom de domaine
    char Fichier[80] ;  // Fichier de type db
    char HV[80] ;       // champ perso à remplacer par vos initiales
};
```

Modifier la fonction **list_print** pour qu'elle fasse les nouveaux affichages.

Soit le fichier named.conf suivant .

On demande de rajouter dans la liste une structure zone pour chaque entrée zone de ce fichier
Vous initialisez les champs de la structure zone en faisant des strncpy.

```
acl internal { 127.0.0.1 ; 192.168.189/24 ; };

options {
    directory "/etc/bind/";
    allow-query { internal ; };
#    forwarders { 10.0.0.1 ; };
    listen-on port 53 { 127.0.0.1 ; 192.168.189.128 ; };
    pid-file none ;
#    include "/etc/named.d/forwarders.conf";
#    include "/etc/named.d/forwarders.conf";
#    query-source address * port 53;
#    cleaning-interval 120 ;
#    statistics-interval 0;
#    notify no ;
};

zone "world.cosmos" in {
    type master;
    file "world.zone";
    HV "HERMAN VANSTAPEL" ;
    allow-update { internal ; };
    allow-transfer { 192.168.1.20 ; };
};

zone "113.112.111.in-addr.arpa" in {
    type master;
    file "111.112.113.zone" ;
    HV "HERMAN VANSTAPEL" ;
    allow-transfer { 192.168.113.129 ; };
};
```

```

zone "114.112.111.in-addr.arpa" in {
    type master;
    file "111.112.114.zone" ;
    HV "HERMAN VANSTAPEL" ;
    allow-transfer { 192.168.113.129 ; } ;
};

zone "localhost" in {
    type master;
    file "localhost.zone" ;
    HV "HERMAN VANSTAPEL" ;
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone" ;
    HV "HERMAN VANSTAPEL" ;
};

zone "." in {
    type hint;
    file "root.hint" ;
    HV "HERMAN VANSTAPEL" ;
};

```

Le résultat du programme doit donner ceci

```

vanstapel@vanstap2:~/lib2022G/D5$ ./main
-- Initialisation --
Taille 0

-- Insertion --
La taille est 1
Taille 1
world.cosmos Fichier:world.zone Herman Vanstapel

La taille est 2
Taille 2
world.cosmos Fichier:world.zone Herman Vanstapel
113.112.111.in-addr.arpa Fichier:111.112.113.zone Herman Vanstapel

```

5) Ecrire un programme qui lit le fichier named.conf de la page 9 et à l'aide du sscanf recherche tout les éléments zone pour ensuite les insérer dans la liste p_sll.
Le programme terminera par un affichage

Le codage

Le programme comportera la fonction suivante

void LectureNamedConf(char *Nom, sll_s * p_sll)

Nom : le fichier named.conf

p_sll : La liste de zones

Faire un fgets pour lire ligne par ligne

Repérer chaque zone

Pour chaque zone, il faut ensuite récupérer le nom de fichier

La valeur associée au record hv , (remplacer hv par vos initiales)

L'usage du sscanf est obligatoire

On peut considérer que la lecture du **};** entrainera l'écriture du record de la zone avec son fichier associé et la valeur de hv en cours dans la liste sll.

Exemple de code

```
FlagZone = 0;
while ( fgets(Buffer,sizeof Buffer,fp),!feof(fp) )
{
    bzero(B1,sizeof B1 ) ; bzero(B2,sizeof B2 ) ; bzero(B3,sizeof B3 ) ;
    sscanf(Buffer,"%30s %30s ",B1,B2) ;

    if (strcmp(B1,"zone") == 0 )
    {
        FlagZone = 1 ;
        EnleveGL(B2) ;
        // A completer
    }
    /// A completer
    else
    if (strcmp(B1,"};" ) == 0 && FlagZone)
    {
        printf(">}; %s\n", B2 ) ;
        FlagZone = 0 ;
        // A completer
    }
    else
        printf("# %s",Buffer) ;
}
```

On utilisera la fonction **list_print** de l'étape précédente pour afficher la liste

Exemple d'exécution

```
-- Initialisation --
# acl internal { 127.0.0.1 ; 192.168.189/24 ; };
#
# options {
#     directory "/etc/bind/";
#     allow-query { internal ; };
# #     forwarders { 10.0.0.1 ; } ;
#     listen-on port 53 { 127.0.0.1 ; 192.168.189.128 ; };
#     pid-file none ;
# #     include "/etc/named.d/forwarders.conf";
# #     include "/etc/named.d/forwarders.conf";
# #     query-source address * port 53;
# #     cleaning-interval 120 ;
# #     statistics-interval 0;
# #     notify no ;
# };
>zone world.cosmos
#     type master;
>Fichier world.zone
>HV HERMAN
#     allow-update { internal ; } ;
#     allow-transfer { 192.168.1.20 ; } ;
>};
#
>zone 113.112.111.in-addr.arpa
#     type master;
>Fichier 111.112.113.zone
>HV HERMAN
#     allow-transfer { 192.168.113.129 ; } ;
>};
#
#
>zone 114.112.111.in-addr.arpa
#     type master;
>Fichier 111.112.114.zone
>HV HERMAN
#     allow-transfer { 192.168.113.129 ; } ;
>};
#
#
>zone localhost
#     type master;
>Fichier localhost.zone
>HV HERMAN
>};
#
#
>zone 0.0.127.in-addr.arpa
#     type master;
```

```

>Fichier 127.0.0.zone
>HV HERMAN
>};
#
#
>zone .
#           type hint;
>Fichier root.hint
>HV HERMAN
>};
#
# #    logging {
# #           category default { log_file; };
# #           channel log_file { file "error" size 10M; };
# #    };
#####
Taille 6
world.cosmos Fichier:world.zone HERMAN
113.112.111.in-addr.arpa Fichier:111.112.113.zone HERMAN
114.112.111.in-addr.arpa Fichier:111.112.114.zone HERMAN
localhost Fichier:localhost.zone HERMAN
0.0.127.in-addr.arpa Fichier:127.0.0.zone HERMAN
. Fichier:root.hint HERMAN

```

6) Expliquer l'exemple suivant

```
int main (void)
{
    char *pchaine =NULL ;
    char Entree[80] ;
    int Octets[4] ;
    // sll_s *p_sll = NULL;
    // p_sll = sll_new () ;
    // LectureNamedConf("named.conf",p_sll) ;
    // printf("#####\n") ;
    // list_print(p_sll) ;

    printf("-----\n") ;
    printf("Entrez une ip ou un nom de domaine;") ;
    fgets(Entree, sizeof(Entree) , stdin ) ;
    Enleve1013(Entree) ;
    fprintf(stderr,"#%s#\n",Entree) ;
    if (Ipv4ToArray(Entree,Octets)!=-1)
    {
        printf("Recherche Netid %d.%d.%d\n",Octets[0],Octets[1],Octets[2]) ;
        // pchaine= RechercheNetID(Octets, p_sll) ;
    }
    else
    {
        char *Hote ;
        char *Domaine ;
        Hote=SepareHoteDomaine(Entree, &Domaine) ;
        printf("Hote:%s Domaine:%s\n",Hote,Domaine) ;
        // pchaine= RechercheNomDomaine(Domaine, p_sll) ;
    }
    if ( pchaine!=NULL )
        printf(" Resultat %s\n",pchaine ) ;
    else
        printf("Rien trouvé !\n") ;
}
```

7) On vous demande maintenant d'écrire une des fonctions de recherche manquantes de l'exemple précédent à savoir

```
char* RechercheNomDomaine(char *Domaine, sll_s * p_l) ;
```

Voici un exemple d'exécution avec un nom de domaine, n'oubliez pas d'ajouter un nom d'hôte au début, vous devrez remplacer par votre domaine personnel

```
#####  
Taille 6  
world.cosmos Fichier:world.zone HERMAN  
113.112.111.in-addr.arpa Fichier:111.112.113.zone HERMAN  
114.112.111.in-addr.arpa Fichier:111.112.114.zone HERMAN  
localhost Fichier:localhost.zone HERMAN  
0.0.127.in-addr.arpa Fichier:127.0.0.zone HERMAN  
. Fichier:root.hint HERMAN  
-----  
Entrez une ip ou un nom de domaine;toto.world.cosmos  
#toto.world.cosmos#  
Hote:toto Domaine:world.cosmos  
Domaine #world.cosmos# zone #world.cosmos#  
Trouvé  
Resultat world.zone
```

8) On demande maintenant d'écrire deux programmes réseaux à l'aide de la librairie lib2022

Un client *nslookup*

Un serveur qui va utiliser la librairie dnslib pour accéder aux fichiers de configuration

Le client va vous demander d'encoder au choix un nom de domaine FQDN ou une ip.

Le serveur va en déduire le nom de domaine ou le netid et va retourner le nom de fichier associé. Le serveur va se servir des fonctions suivantes écrites à l'étape précédente

```
LectureNamedConfHV      avec sscanf
list_namedconfHV(p_sll)
RechercheNomDomaineHV
```

Voici la copie de l'écran du client en cas de requête sur un nom de domaine

```
vanstapel@gateway:~/lib2021/ndns4 Recherche en Réseau$ ./cli 127.0.0.1 1400 127.0.0.1 1300
port 1400
CreateSockets 3
Entrez une ip ou un nom de domaine;toto.world.cosmos
#toto.world.cosmos#
Envoi de 44 bytes
bytes:44:world.zone
```

Voici l'affichage du serveur

```
vanstapel@gateway:~/lib2021/ndns4 Recherche en Réseau$ ./ser 127.0.0.1 1300
Ceci est le serveur
port 1300
CreateSockets 3
Ouverture réussie de named.conf
#####
Taille 6
world.cosmos Fichier:world.zone HERMAN
113.112.111.in-addr.arpa Fichier:111.112.113.zone HERMAN
114.112.111.in-addr.arpa Fichier:111.112.114.zone HERMAN
localhost Fichier:localhost.zone HERMAN
0.0.127.in-addr.arpa Fichier:127.0.0.zone HERMAN
. Fichier:root.hint HERMAN
-----
bytes:44:toto.world.cosmos
Hote:toto Domaine:world.cosmos
Domaine #world.cosmos# zone #world.cosmos#
Trouvé
Resultat world.zone
bytes:44
```


9) Lecture du fichier world.zone

on vous demande maintenant d'écrire un programme *nslookup* qui lit les records d'un fichier de db.domain, par exemple world.zone .

Ecrire la fonction suivante à ajouter dans dnslib:

void LectureDBHV(char *NomFichier,sll_s *pl) ;

Cette fonction en utilisant la fonction **parse** va créer une liste sll_s composée des structures EntreeZone dont vous avez la description ci-dessous.

```
enum TypeIN { NS=1, A=2 , PTR=3 ; HV=4 } ; # le type HV est à remplacer par vos initiales

struct EntreeZone
{
    char Nom[80] ;
    char Valeur[80] ;
    enum TypeIN Type ;
};
```

Ecrire ensuite la fonction suivante pour afficher le contenu de cette sll_s

void list_DBHV(sll_s *p_l) ;

Enfin une fonction de recherche dans la sll_s

char *Recherche_DBHV(char *Nom,int Type,sll_s *p_l) ;

Le programme de Test devra donner le résultat suivant

```
vanstapel@gateway:~/lib2021/ndns5 worldeone$ ./main
V1.0 Ouverture réussie de world.zone
// Pour plus de clarté , les informations de parsing de world.zone ont été supprimées

-----
Debut 1.1
earth 2 IN A 111.112.114.2
europe 1 IN NS tati.europe
tati 2 IN A 192.168.113.131
gateway 2 IN A 111.112.113.114
mars 2 IN A 111.112.113.200
Maxime 2 IN A 111.1.1.1
moon 2 IN A 111.112.113.3
nono 2 IN A 111.112.113.200
HV 4 IN HV"HERMANVANSTAPEL"
sun 2 IN A 111.112.113.2

>set type=A
Le type choisit est 2
>mars
#mars#
111.112.113.200
```

```
>set type=HV  
Le type choisit est 4  
>HV  
#HV#  
"HERMANVANSTAPEL"  
>
```

10) Intégration des point 11 dans le point 12. Le serveur doit répondre à une requête type nslookup depuis un client

Le client doit en premier demander sur quel type de record porte la requête

Le serveur quand il reçoit une requête pour un nom de domaine (FQDN) doit ouvrir le fichier correspondant suite à un appel à RechercheNomDomaine.

Charger ce fichier grâce à la fonction LectureDB

Séparer le Hôte du FQDN

Faire une recherche sur le Hôte via la fonction Recherche_DBHV

Retourne l'info obtenue au client, typiquement une IP

Si une IP est envoyée au serveur, le serveur la reconnaît bien mais se contente de retourner simplement "Recherche Netid non gérée" au client.

Voici l'écran du client:

```
vanstapel@gateway:~/lib2021/ndns6$ ./cli 127.0.0.1 1400 127.0.0.1 1300
port 1400
CreateSockets 3
>set type=A
Le type choisit est 2
>mars.world.cosmos
#mars.world.cosmos#
Envoi de 44 bytes
bytes:44:111.112.113.200
>set type=HV
Le type choisit est 4
>HV.world.cosmos
#HV.world.cosmos#
Envoi de 44 bytes
bytes:44:"HERMANVANSTAPEL"
```

11) Intégrer un timer à l'exercice 12

Si le programme nslookup ne reçoit pas de réponse après 4 secondes, afficher timeout !!! et retransmettre bien entendu la requête jusqu'à ce que le serveur la traite correctement

12) Programmer la Délégation

Créer un sous-domaine de votre domaine et faire en sorte que les requêtes concernant votre nouveau sous-domaine soit adressées au serveur de votre domaine soit déléguée au serveur du sous-domaine. Voir la syntaxe au cours. Vous devez utiliser le record NS et ensuite lire l'IP du second serveur named.

Le serveur du sous-domaine doit répondre au serveur du domaine principal et ce dernier répondra enfin au client

Exemple : Vous avez deux domaines Villes.com et Belgique.Villes.com

Une question concernant paris.Villes.com doit être gérée par le serveur responsable de Villes.com.

Une question sur Liege.Belgique.Villes.com sera déléguée par le serveur de villes.com vers le serveur de Belgique.Villes.com.

Déléguer signifie que toute requête est toujours transmise au serveur de villes.com par nslookup mais si named se rend compte que la requête concerne en fait le sous-domaine Belgique, il envoie la requête au serveur named responsable de ce sous-domaine.

Belgique.Villes.com retournera la réponse vers villes.com qui retournera la réponse au client.

Nslookup enverra toujours les questions vers le serveur de Villes.com qui déléguera si nécessaire.

Attention ; Si je tape n'importe quoi comme sous-domaine, le programme doit le signaler. Si je tape un nom de domaine France.Villes.com. Ce nom doit être signalé comme invalide.