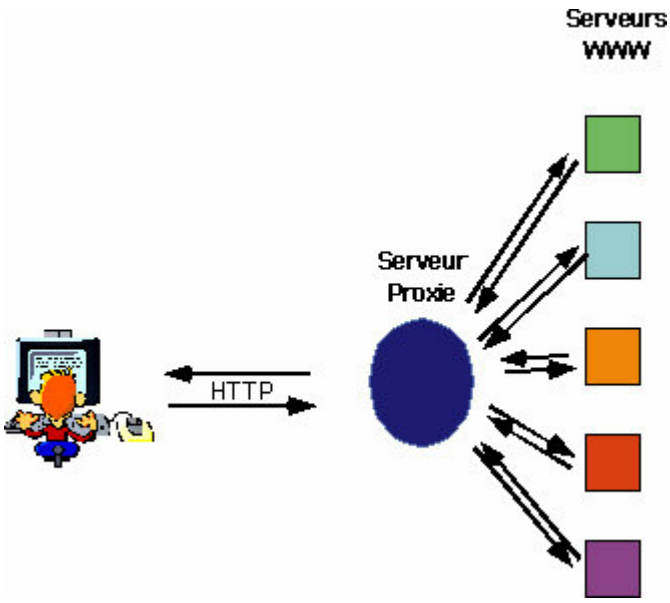
LUCCHI Aurélien

Filtre du Web avec contrôle parental

I. Principe

Notre filtre du Web fonctionne en fait comme un serveur proxy.



Un serveur proxy est une entité intermédiaire entre le client et le serveur ; il a en fait de multiples fonctions, notamment l'interrogation du serveur à la place du client puis la redirection des informations vers le client (une adresse ip et un numéro port, tous deux mémorisés dans une socket).
Dans notre cas –et c'est là son rôle essentiel- le serveur proxy effectue aussi un contrôle du contenu de la page demandée. En effet, si la page demandée contient des mots qui doivent être refusés (ces mots sont saisis par l'utilisateur du proxy) et que les conditions de refus sont remplies (déterminées selon un algorithme présenté plus bas), alors la page demandée n'est pas retournée. A la place, on retourne une page précisant que la page initialement demandée a été refusée. Ceci permettra par exemple un contrôle parental en vu d'éviter à des enfants de surfer sur des pages à caractères pédophiles ou racistes.

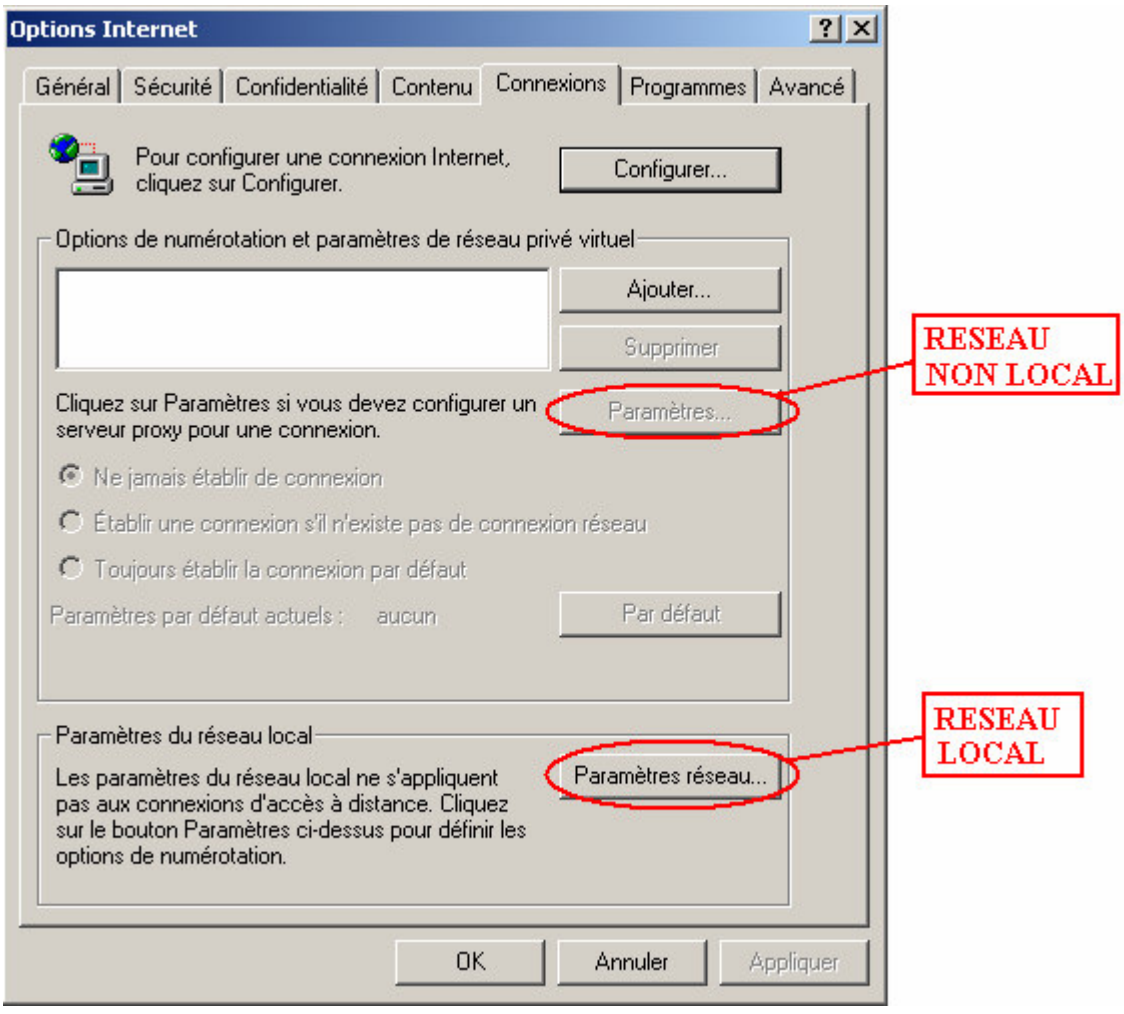
II. Choix de Windows

Ceci a surtout permis de voir l'utilisation des sockets sous Windows qui est légèrement différente de celle sous Linux ou Unix. Si les fonctions utilisées restent les mêmes, il est nécessaire d'effectuer une opération supplémentaire : l'initialisation de Winsock, une DLL sous Windows qui permet d'utiliser des sockets TCP/IP. Cette étape est **indispensable**, sans quoi Windows ne renverra aucune socket valide au programme.

Les threads
Un thread (appelé aussi processus léger ou activité) est « un fil d'instructions (un chemin d'exécution) à l'intérieur d'un processus ». Contrairement aux processus, les threads d'un même processus partagent le même espace d'adressage, le même environnement (par exemple les mêmes variables d'environnement, des mêmes données, etc.). Ainsi deux threads d'un même processus communiquent beaucoup plus facilement que 2 processus sous Windows. Le système d'exploitation (sous les systèmes monoprocesseur) partageant le temps processeur entre les différents threads (en donnant régulièrement accès à chacun d'eux au processeur pendant un très court instant), cela peut être utile pour exécuter des processus lents sans bloquer l'application.

La classe TThread
Pour des questions de maintenance du code, le choix s'est porté la classe TThread de la VCL de C++Builder. De plus, cette classe permet de gérer facilement l'accès à des composants de notre fenêtre, des variables ou des zones mémoires. En effet, il est nécessaire de synchroniser les threads qui pourraient, dans le cas inverse, accéder en même temps à une ressource. Cette synchronisation est implémentée de manière plus transparente par la classe TThread et le code est donc plus facile à lire.

III. Communication entre le client et l'application



Cette utilisation peut être demandée par le client ou être faite de manière automatique lors du lancement du programme à condition que l'utilisateur possède le navigateur Internet Explorer. En général, le paramétrage se fait dans le menu "outils" ou "options internet" du navigateur, puis dans la rubrique "paramètres" (ch schéma).

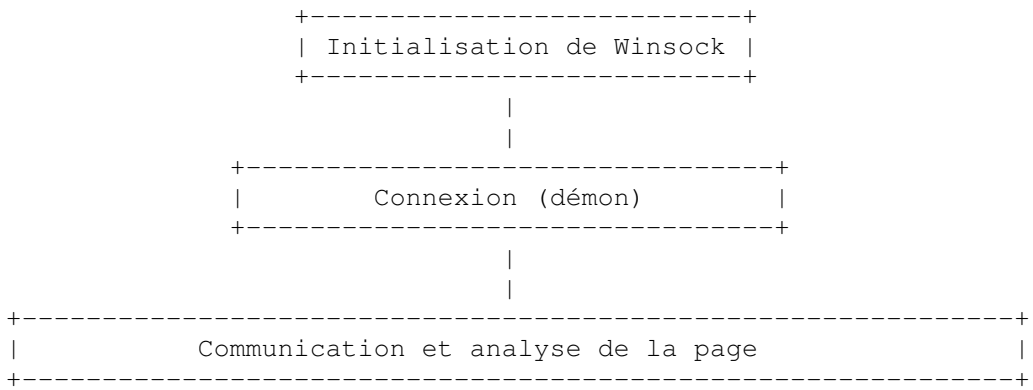
Configuration manuelle
Il faut indiquer le nom et le numéro de port du proxy pour chaque protocole. Dans notre cas, le proxy est lancé sur notre propre machine, il faut donc indiquer au navigateur notre propre adresse ip, soit 127.0.0.1.
Le port sur lequel fonctionne notre proxy est paramétrable (8010 par défaut).

Inconvénient
Ceci demande l'intervention de l'utilisateur.

Avantage
Notre proxy peut très bien être lancé d'un autre pc, une passerelle par exemple. Ainsi, notre logiciel n'est pas restreint au simple cas du pc personnel familial, mais il pourra aussi être utilisé en entreprise.

Configuration automatique
Etant donnée la multiplicité des navigateurs existant, il n'a pas été possible de fournir une procédure automatique pour chaque navigateur. Il y a tout de même une procédure automatique pour le navigateur Internet explorer qui reste le plus utilisé des navigateurs Internet sous Windows (sûrement pas le meilleur, comme le prouve Firefox…).

IV. Détails des fonctions



Init() : Cette fonction est importante dans le bon fonctionnement de notre serveur proxy. En effet, est sert à :
- L'initialisation de WinSock. Cette étape est INDISPENSABLE, sans quoi le Windows ne renverra aucune socket valide au programme.
- La mise sur écoute de notre serveur au port choisi.

Winsock : Windows Sockets. Une fameuse DLL sous Windows qui permet d'utiliser des sockets TCP/IP.

Connexion() : Cette fonction sera lancée comme **thread** qui fonctionne en permanence en tâche de fond, attendant d'être sollicité pour répondre à une requête. C'est en fait un daemon

(démon en français) implémenté grâce à une simple boucle infinie (while(1)…). Cette fonction sera en permanence à la recherche de nouveaux clients désirant obtenir une page web.

Communication() est aussi une fonction lancée en tant que **thread** par la méthode Connexion() à chaque fois qu'un client fait une nouvelle requête au serveur proxy. Communication() sera chargée de dialoguer avec le serveur contenant la page que le client a demandé pour déterminer si oui ou non, cette page doit être acceptée. L'analyse de la page sera confiée à la méthode Analyse(). La réception sera elle confiée à une autre méthode : RéceptionPage().

Analyse() : Cette fonction retourne un booléen indiquant si la page doit être ou non acceptée. Pour cela, l'utilisateur indique des mots qu'il souhaite interdire, ainsi qu'un « poids » associé à chaque mot. Le poids d'un mot est en fait un entier compris entre 1 et 10 indiquant le nombre de fois ou l'on peut accepter un mot dans une page. Par exemple, le mot « prison » avec un poids de 1. Dans ce cas, si la page contient une seule fois le mot « prison », elle sera refusée. Cet algorithme simple à mettre en place aura l'avantage de ne pas trop consommer de temps CPU pendant la navigation de l'utilisateur.

Remarques :
Il y a une fonction permettant la réduction du programme dans la trayicon de Windows (barre des tâches à droite). Ainsi, le programme ne gênera en rien le surf d'un utilisateur. D'autres options ont été implémentées afin de fournir à l'utilisateur un programme plus convivial et facile d'accès. De ce point de vue, il aurait peut être été plus aisé de fournir une procédure d'automatisation générique (pour tous les navigateurs). Ceci semblait techniquement possible mais inadapté au cahier des charges qui demandait simplement un filtre du Web (fonctionnant habituellement sur le port 80) et non de tous les protocoles (messageries, chat,…). Il aurait alors fallu analyser toutes les trames entrantes, ce qui aurait consommé plus de ressources CPU…

De plus, puisqu'il s'agissait d'un logiciel de contrôle parental, il a été décidé d'ajouter une fonctionnalité permettant à une personne (un parent par exemple) de bloquer le logiciel avec un mot de passe. Le mot de passe a été crypté par un algorithme plutôt simple appelé XOR. Cela revient en fait à inverser ou non certains bits du message en sachant ré-inverser ceux qui ont été inversés. En recryptant, on retombe sur ses pieds d'où ce nom de XOR. Voilà donc pourquoi il a été préféré cet algorithme à un autre (RSA, MD5,…) car il est beaucoup plus simple à mettre en oeuvre et évite d'augmenter de manière trop importante la taille du programme (la dll pour le cryptage md5 pesant plus de 300Ko, sans compter les autres librairies à ajouter.).

V. Utilisation d'un cache

Afin de gagner du temps lors de l'analyse des pages Web (qui peut consommer un temps CPU non négligeable sur des pc assez anciens surtout), un cache à été mis en place. Il s'agit en fait d'un fichier log contenant toutes les adresses visitées avec la date et l'heure de visite (mis à jour à chaque visite sur le site). Lorsque l'on effectue une lecture dans le fichier et que l'on y trouve l'adresse recherchée, on doit aussi regarder la valeur de la date et de l'heure inscrite dans le fichier et les comparer à la date et à l'heure de la dernière modification de la liste des mots interdits. En effet, si le site est présent dans le cache mais que l'on a modifié la liste de mots depuis la dernière visite, il est alors nécessaire d'analyser de nouveau le site en entier pour satisfaire aux nouvelles conditions imposées. La date de modification de la liste des mots devra être stockée dans un endroit qui permet sa récupération à n'importe quel moment. Pour cela, on utilise la base de registre Windows.

Voici un exemple :
Un utilisateur a visité le site www.siteinterdit.com le 15/10/03 à 09h34m10s. La dernière modification de la liste des mots ayant été effectué le 12/10/03 14h52m14s, le statut présent dans le fichier est alors valide.

VI. Mutex pour gestion accès écriture au fichier log

Un "mutex" permet de gérer des exclusions mutuelles et permet de protéger des données, des zones d'exécution et de synchroniser des tâches. Il possède deux états, "unlocked" c'est à dire qu'il n'est pas attribué à une tâche donnée, ou "locked" c'est à dire qu'il appartient à une tâche. Une tâche qui tente de verrouiller un mutex pris par une autre tâche est suspendue jusqu'à la libération du mutex par son propriétaire.

Utilisation :

```
HANDLE mut;
mut=CreateMutex(NULL,FALSE,"mutex");
CloseHandle(mut);
…
WaitForSingleObject(mut,INFINITE);
Travail();
ReleaseMutex(mut);
```

VII. Quelques options

Il y a quelques options supplémentaires à ce logiciel comme la quantité de mémoire que le programme peut utiliser, la possibilité d'activer le logiciel dès le démarrage ou encore l'utilisation d'un serveur proxy (le logiciel enverra alors les requêtes au serveur spécifié et non à la page concerné) avec tous les avantages (anonymat, contrôle du trafic) et les inconvénients (ralentissement possible) que cela entraîne.

Ce projet utilise également quelques contrôles graphiques fournis par une bibliothèque : rxlib. Des détails sur cette bibliothèque peuvent être trouvé ici :
http://ricky81.developpez.com/tutoriel/bcb/rxlib/install/

VIII. Conclusion

Ce projet était intéressant car il a permis de mettre en pratique les techniques vues en cours de réseaux et de programmation et de les approfondir dans le cadre d'un vrai projet.

Evolution

On aurait pu imaginer la mise en place de plusieurs profils d'utilisateurs (donc création de plusieurs listes de mots à interdire, une pour chaque utilisateur).
La création de listes noires personnelles d'URL et d'une liste blanche aurait également pu encore améliorer les performances…