TP sur les protocoles internet

Loris Croce

1. — `printf` : 3
   — `pwd` : 1
   — `read` : 1, 2
2. Dans les sections 1 et 2.
3. `printf` est issue d'une bibliothèque tandis que `write` est une instruction sytème.

Programmation *socket*

Programmation d'un client/serveur UDP

4. cf `tp4serveur.c`
5. *"The htons() function converts the unsigned short integer hostshort from host byte order to network byte order."* Il permet de convertir les valeurs *hôtes* vers des valeurs *réseaux*.
6.
```
$  ./tp4 10001
écoute...
udp        0      0 0.0.0.0:10001                   0.0.0.0:*
    5255/./tp4
```

Client UDP

7. Oui.
8. cf `tp4client.c`
9. Il faudrait remplacer l'instruction `fgets()` par `scanf()`.

Communication UDP

10. (Personne de disponible).
11. Non il ne semble pas adapté car il utilise un mode de transmission *sans connexion*. Ce qui ne permet pas de répondre aux prérequis.

Programmation d'un client/serveur TCP

Serveur TCP

12. Il doit en gérer 3.
13. (...)
14. (...)
15. Car dans TCP le socket lève une exception quand le client se coupe.
16. (...)

Client TCP (...)

Programmation de protocoles Internet

21. — `200` : OK
    — `301` : Redirection permanente
    — `302` : Redirection temporaire
    — `400` : Requête incorrecte
    — `403` : Accès interdit (si connection les droits sont insuffisants)
    — `404` : Ressource introuvable
    — `418` : *Je suis une théière* (easter egg)
    — `500` : Erreur interne du serveur
22. `$ nslookup www.google.com` pour avoir l'IP de l'URL. puis `$ telnet 216.58.205.4 80` qui renvoie du code html dans stdout (cf `google.html`).
23. cf `http.c`.

(...)