NE302 : Projet HTTP server.

- Administratif:
  - Enseignant:
    - Christophe Deleuze,
    - Quentin Giorgi,
    - Yves Guido.
  - 3 ECTS (soit plus de 80h de travail en tout)

    http://ec.europa.eu/education/ects/users-guide/assets/ects_users-_guide_web.pdf
  - Evaluation:
    - N1=CC
      - ((QCM*15)+(EVAL1*35)+(EVAL2*50))/100)
    - N2=N1 (pas de rattrapage)

NE302 : Projet HTTP server.

- Objectifs pédagogiques :
  - Etre capable de comprendre une spécification (RFC), d'en extraire les informations essentielles, mais aussi de rechercher les détails nécessaires à l'implémentation.
  - Plus spécifiquement, comprendre le protocole et les mécanismes mis en œuvre dans une architecture HTTP. (client / serveur, applications et suivi de sessions
  - Etablir les liens avec les autres cours (informatique théorique : notion de grammaire) et algorithmique (recherche dans des structures de données)
  - Réaliser l'implémentation de la spécification, en utilisant des méthodes et outils de développement reconnus.
  - Réaliser un travail de groupe (en trinome, 2 groupes de 8 trinomes) en respectant délais et objectifs.

NE302 : Projet HTTP server.

- Ce que le projet n'est pas :
  - Le développement d'une application web complexe (PHP / python / ruby /etc..)
  - Un projet de programmation de l'API des socket réseau
    - la connaissance de l'API des sockets n'est pas strictement nécessaire, et sera masquée par une bibliothèque fournie.
  - Un projet système (communication inter processus, multi threading, optimisation de la mémoire, etc...)
    - en ce sens le projet n'a pas vocation à un exemple d'architecture logicielle d'un serveur, mais permettra tout de même aux étudiants de se poser les bonnes questions sans avoir nécessairement les pré-requis pour avoir dès maintenant les réponses.

NE302 : Projet HTTP server.

- Phase du projet :
  - *Etape 0 : Présentation du projet*
    - *1 séance de 30 minutes en commun*
  - *Etape 1 : Comprehension des spécifications.*
    - Fourniture : RFC et passages importants indiqués
    - 1 séance de 3h sur machine: Découverte ou rappel de HTTP.
      - 1 enseignant par groupe.
    - Evaluation : QCM (45 minutes max) sur les RFCs le 15/02.
      - Documents papier autorisés.
  - *Etape 2 : Développement d'un parseur de requêtes HTTP*
    - Fourniture : une API à respecter, et un jeu de tests.
    - 2 séances de 1h30 de tutorat/travail en salle machine.
      - 2 enseignants par groupe.
    - Evaluation : Rendu du travail (code) le 15/03 sur chamilo.

NE302 : Projet HTTP server.

- Phase du projet :
  - *Etape 3 : Prise en compte des requêtes pour servir des fichiers locaux :*
    - Fourniture : une API accès réseau, et un jeu de tests de requêtes.
    - 3+1 séances de 1h30 de tutorat/évaluation – 1 toutes les 2 semaines :
      - 2 enseignants par groupe.
      - Formaliser les points d'avancement :
        - rendu du travail intermédiaire archivé la veille de la séance.
        - démonstration avec ce travail lors de la séance.
    - Attendu fonctionnel:
      - Gestion de la réception des requêtes, vérification syntaxique et sémantique
      - Gestion des entêtes HTTP
      - Normalisation de l'URL
      - Gestion de l'accès aux fichiers et type mime (basé sur l'extension) et le charset par défaut, gestion multi-sites.
      - Optionnel : Gestion des encodages : Chunked / Deflate / gzip
    - Evaluation : Rendu du travail le 01/05 sur chamilo.
      - Démonstration avec des fichiers statiques et des fichiers javascript sur des pages web.

NE302 : Projet HTTP server.

- Phase du projet : fin étape 3 et étape 4
  - *Etape 4 : Prise en compte des requêtes pour servir des applications*
    - Fourniture : un squelette de code pour les accès réseaux.
    - Une semaine bloquée fin juin après les examens
      - 5 jours, la dernière demi journée est pour l'évaluation
      - Permanence d'un enseignant chaque jour sauf pour l'éval (3 enseignants).
    - Attendu fonctionnel :
      - Gestion de l'interface fastCGI :
      - Gestion des réceptions fastCGI/emission HTTP.
      - Démonstration avec un serveur d'application PHP
    - Livraison/évaluation :
      - Démonstrateur fonctionnel
      - Indicateur de qualité du code source.
      - Preuve d'execution. (screenshot / tcpdump)
      - Soutenance finale.