**Beoordelingscriteria**

**De server functioneel**

* Wacht op een tcp-poort op verbinding van een client
* Leest opdracht, handelt die af en stuurt informatie terug naar client
* Houdt verbinding open totdat de client opdracht tot sluiten geeft
* Draait op de achtergrond (heeft geen user-interface)
* Hoeft niet multithreaded
* De server is niet door een client af te sluiten.

**De client functioneel**

* Heeft een user-interface (tekst of grafisch)
* Krijgt de naam of het ip-adres van de gewenste server van de gebruiker
* Maakt verbinding met server als de gebruiker daar om vraagt.
* Stuurt namens de gebruiker opdrachten naar de server
* Verwerkt de teruggestuurde resultaten en laat informatie zien aan de gebruiker.

**Het protocol functioneel**

* Het is een applicatieprotocol op tcp.
* Het protocol is afgestemd met een andere groep.
* Er is een duidelijke beschrijving van het protocol.
* De volgende opdrachten zijn in iedere geval geïmplementeerd:
  + INFO geeft naam en versie van de server
  + DIR *dirname* geeft een lijst van bestandsnamen in een directory, bijvoorbeeld DIR C:\TEST
  + GET om een bestand van de server op te halen
  + PUT om een bestand naar de server te sturen
  + DEL om een bestand op de server te verwijderen
  + REN om een bestand op de server te hernoemen
  + QUIT sluit de verbinding
* Alle communicatie bestaat uit leesbare regels tekst, afgesloten met een newline ('\n') en geen carriage return ('\r')
* Antwoorden van de server kunnen uit meerdere, niet-lege regels bestaan
* Aan het eind van een antwoord stuurt de server altijd een lege regel
* Bestandnamen met spaties en/of niet-ascii tekens geven geven geen problemen, bijvoorbeeld "Mijn ideeën".
* Client en server kunnen elkaar niet laten vastlopen.
* Storing of wegvallen van de verbinding moet geen vastlopers veroorzaken.
* Binaire bestanden zoals .exe en .jpg komen onbeschadigd over.
* Een bestandsoverdracht kan onderbroken worden (bijvoorbeeld met ctrl-c). Daarbij blijven geen incomplete bestanden achter.
* *Busy waiting* en *polling* zijn zo veel mogelijk vermeden.
* Het belangrijkste commando is natuurlijk:
  + SYNC *local-folder remote-folder*  
    Dit commando zorgt dat een map op de client (local-folder) wordt gesynchroniseerd met een map op de server (remote-folder). Met andere woorden: alle wijzigingen in die in de map (en submappen) aan de client-kant zijn aangebracht, worden gespiegeld op de aangegeven map (en submappen) op de server. Volg de volgende regels:
    - Bestanden die niet bestaan op de client en wel op de server: worden verwijderd
    - Bestanden die bestaan op client maar niet op de server: worden naar de server verzonden
    - Bestanden die nieuwer zijn op de client dan op de server: worden naar de server verzonden
    - Zorg ervoor dat bestanden bij de volgende SYNC niet onnodig opnieuw worden verzonden

**C++**

Je schrijft een client én een server. Deze moeten met elkaar kunnen communiceren. De server moet in *unmanaged* C++ geschreven zijn. De client mag eventueel ook in managed C++, C# of Java gebouwd worden.

* Er is een duidelijke verdeling in klasses
* Namen van klasses, typen, variabelen en functies zijn logisch, duidelijk en consequent
* Elke klasse heeft een .ccp en een .h
* De code is voorzien van zinvol commentaar
* In elke .h-file worden de niet-private onderdelen van de klasse beschreven voor de programmeur.
* Er zijn aantoonbaar geen memory leaks
* Foutafhandeling werkt met try-catch
* De demo-versie is *zonder debugging* gebouwd
* Bestanden worden verstuurd in blokken van eeen paar KB, om snel te zijn en toch zuinig met geheugen.

**Algemene eisen**

* Er is gewerkt in een tweetal
* Beide teamgenoten begrijpen alle code en kunnen de code en keuzes verklaren.
* De demo is goed voorbereid. Voorkom het demonstratie-effect ("gisteren werkte het nog");
* Alle functionele eisen worden gedemonstreerd.
* De demo moet zonder visual studio gegeven kunnen worden.
* Je kunt demonstreren dat je met client en server van een ander team kunt communiceren

|  |
| --- |
| **Voorbereiding** |
|  |
| **Zorg ervoor dat de volgende zaken geregeld zijn voordat je bij 't assessment binnenkomt.** |
|  |
| je hebt een laptop bij je |
| er staan twee werkende projecten op voor de server, waar we ter plekke in de code kunnen duiken: één  voor Windows en één voor Linux of Mac OS X |
| je hebt een losse release-build bij je van de server: een aparte executable, die zonder ontwikkelomgeving  is op te starten (bij voorkeur de Windows-versie) |
| je hebt een werkend client-programma (mag vanuit ontwikkelomgeving, maar mag ook als losse  executable) |
| op de machine waar je client draait heb je de test-directory staan die we op Blackboard klaar hebben  gezet (hierin zitten diverse soorten bestanden, soms met rare namen, zodat we de overdacht van  bestandsnaam en -inhoud goed kunnen testen) |
|  |

|  |
| --- |
| **Beoordeling** |
|  |
| **De volgende zaken *moeten* in orde zijn, wil je een voldoende kunnen halen;**  **niet in orde = geen punt = herkansing!** |
|  |
| *gedetailleerde* protocolbeschrijving beschikbaar |
| communicatie tussen client en server verloopt volgens protocol (dit moet **exact** kloppen!) |
| niet meer dan 2 memory leaks |
| commando's moeten werkend geïmplementeerd zijn |
| INFO |
| DIR |
| GET |
| PUT |
| DEL |
| REN |
| QUIT |
| SYNC |
| server is beschikbaar voor Windows en voor Linux of Mac OS X |
| release-versie werkt niet |
| debug-versie werkt niet |
|  |
| **Als bovenstaande zaken in orde zijn start je bij een 8; daar kunnen punten af of bij, afhankelijk van of aan onderstaande zaken is voldaan** |
|  |
| **Bonuspunten** |
|  |
| * is je server multithreaded (dat was geen eis), dan kun je 1 punt extra verdienen |
| * heb je een intelligente sync gebouwd, die een rename kan herkennen en vervolgens zelf ook een   rename doet i.p.v. de hele directory of file overkopiëren, kun je tot 2 punten extra verdienen |
| * (kom je boven een 10 uit, ronden we af naar een 10) |
|  |
| **Maluspunten: steeds 0,5 punt eraf indien niet voldaan** |
| * verbinding moet open blijven totdat client opdracht tot sluiten geeft |
| * protocol is afgestemd met andere groep |
| * bestandsnamen met spaties en/of rare tekens geven geen problemen |
| * client en server laten elkaar niet vastlopen |
| * storing of wegvallen verbinding moet geen vastloper veroorzaken |
| * binaire bestanden zoals een .jpg komen onbeschadigd over |
| * een bestandsoverdracht kan worden onderbroken (bv.met ctrl-c); daarbij blijven geen   incomplete bestanden achter |
| * foutafhandeling werkt met try/catch |
| * bestanden worden verstuurd in blokken van een paar kB, om snel te zijn en toch zuinig   met geheugen |