

## LEERTAAK 2: XCHANGE – EMBEDDED C MET EEN OS

### INLEIDING

In de komende twee weken ga je werken aan het verder uitbouwen van een deels gegeven peer to peer file exchange programma genaamd XChange. Het achterliggende idee is dat iedereen complete en incomplete bestanden via internet beschikbaar stelt aan andere gebruikers van het programma. De noviteit van het systeem is juist dat er in het algemeen niet één gebruiker is die het volledige bestand in z'n bezit heeft, maar juist veel gebruikers stukken van een bestand delen. Door het stukje bestand dat je hebt gedownload meteen weer te delen met andere gebruikers bouw je dus veel redundantie in het systeem. Op deze manier heb je minder last van het feit dat gebruikers steeds aan- en afmelden.

De Xchange applicatie is een eenvoudige vorm van BitTorrent. Een korte beschrijving van het BitTorrent protocol is te vinden op [the-bittorrent-protocol](#) en [Wikipedia](#). Bij Xchange zijn er in tegenstelling tot BitTorrent geen metainfo files en ook geen trackers die bijhouden welke peer welke blokken heeft. Wel is er een nameserver waar peers geregistreerd staan. Verder is er geen intelligente strategie bij het binnenhalen van blokken, ze worden gewoon op volgorde binnengehaald.

Jullie moeten de aangeleverde Java code van XChange analyseren, aanvullen en uitbreiden. Deze uitbreiding is een component die niet op de PC draait, maar op je router ! Het programma op de router dien je in C te schrijven.

### DOEL

Het doel van deze leertaak is:

- Het analyseren van een gedistribueerde applicatie.
- Het opstellen van sequence diagrammen voor deze applicatie.
- Het ontwerpen en implementeren van het ontbrekende deel.
- Testen van het geheel.

### CONTEXT

Walter Kwaakstra, een BI-student, en Goliath Langhaar, een TI-student, besluiten samen om een bedrijfje XS op te richten dat netwerkkapparatuur verkoopt. Ze willen graag opvallen door naast de gewone spullen iets extras aan te bieden. Na een avondje brainstormen hebben ze volgende idee.

Peer to peer file exchange programma's worden veel door studenten gebruikt. Als je bijvoorbeeld complete ISO bestanden wilt downloaden komt het vaak voor dat je iets anders gaat doen terwijl je eigenlijk het programma en je computer niet wilt afsluiten. Het zou dan handig zijn als je wel je computer uit kunt zetten, maar je router de bestanden laat downloaden.

Linksys is een fabrikant van onder andere wireless routers. Het is mogelijk om op bepaalde types een Linux distributie te zetten met behoud van de volledige functionaliteit van de router. Dit stelt je in staat om eigen programma's op de router te plaatsen. In dit geval is dat een peer-to-peer file exchange programma dat op verzoek van je PC gaat up- en downloaden. Op het moment dat je PC weer aanzet laat je het file exchange programma op de router stoppen en haal je de gedownloade blokken op en

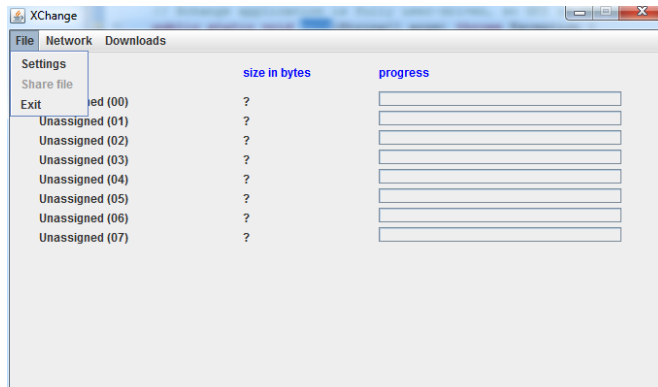
“merge” je deze met de delen die op de PC staan. Probleem is wel dat de router geen harde schijf bevat. Een (externe) storageserver zou uitkomst kunnen bieden.

Helemaal enthousiast door dit concept slaat Goliath aan het programmeren. Echter heeft Goliath last van een korte spanningsboog. Hij vindt het concept wel leuk, maar ontdekt dat hij wel erg veel moet nadenken over protocollen. Hij komt tot het besef dat hij de computer toch altijd aan heeft staan, met als gevolg dat de te bouwen oplossing niet op hem van toepassing is. Na enige woordenwisselingen besluit Goliath zijn samenwerkingsverband met Walter te stoppen.

Walter zit nu met zijn handen in zijn haar. Zoals het een slimme jonge zakenman betaamt heeft hij namelijk de huid van de beer al verkocht voordat deze geschoten is: Yellow, een bekende aanbieder van ADSL vindt het idee van XS leuk, maar eist wel een werkende versie over twee weken.... Het is aan jullie om Walter te helpen.

## BESCHRIJVING FUNCTIONALITEIT XCHANGE

We beschrijven de functionaliteit van de XChange applicatie kort op basis van de bestaande menu layout van de PC-component (RT = router).



### *Settings*

Met deze optie kun je de nameserver, storageserver, router en eigen IP adres invullen. Zonder ingevulde settings kun je verder niets doen.

### *Aan- en afmelden bij nameserver*

Dit gebeurt automatisch bij starten of afsluiten XChange applicatie.

### *List peers*

Met deze optie kun je een lijst opvragen van alle beschikbare (= geregistreerde) peers.

### *File search (add to download list)*

Met deze optie kun je bestanden zoeken bij peers en een bestand toevoegen aan de lijst van bestanden die je wilt downloaden. Je kunt maximaal 8 bestanden tegelijk downloaden.

### *Start PC download*

Met deze optie kun je de download (van internet => PC) starten. Deze optie is alleen beschikbaar als het programma in de toestand "Local Stop" zit.

De filekenmerken (naam, grootte en de progressie in het downloaden) worden in de lijst van maximaal 8 files getoond. Deze 'progress bar' wordt bijgewerkt zodra bij een download een block bij is geplaatst (nog niet volledig geïmplementeerd).

### *Stop PC download*

Met deze optie kun je de PC download stoppen. Deze optie is alleen beschikbaar als het programma in de toestand "Local Start" zit.

### *Start router download*

Met deze optie kun je de download van internet => RT laten beginnen. Deze optie is alleen beschikbaar vanuit de toestand "Local Stop".

Bij de router download worden de bestanden op een storageserver opgeslagen. De router zelf heeft namelijk geen ruimte voor grote bestanden.

#### *Stop router download*

Met deze optie kun je de download van internet => RT stoppen. Deze optie is alleen beschikbaar als het programma in de "RT Start" toestand zit. Alle aanwezige blokken op de storageserver worden gekopieerd naar de PC en 'gemerged' met de op de PC aanwezige blokken.

#### *Exit*

Deze optie zorgt voor een nette afsluiting van het programma en afmelding bij de nameserver.

#### *Locale files*

xchange/settings.txt bevat de settings

De downloads komen terecht in de directory xchange/shared. Hier staan ook de files die je deelt met peers.

Bij het begin van een download wordt al wel de volledige schijfruimte toegewezen, de incomplete file krijgt de extensie .!xch. (Blokken worden dus niet eerst gecashed).

Bij elke download wordt een file met extensie !info in xchange/INFO aangemaakt, hier wordt bijgehouden welke blokken (64 kbytes is default) reeds zijn binnengehaald.

**OPDRACHT 1: [25%] SEQUENCE-, STATE – EN DEPLOYMENT DIAGRAM**

Op blackboard is een eerste werkende implementatie van de XChange PC component gegeven. Bestudeer eerst het protocol in de appendix en bestudeer dan de Java implementatie om een beeld van de software te krijgen.

Use-cases worden meestal als uitgangspunt genomen voor het maken van een sequence diagram. De use cases zijn eerder al globaal geschetst bij de beschrijving van de beschikbare menu opties.

Maak sequence diagrammen voor de volledige set van Use Cases van de XChange applicatie. Welke entiteiten (UML : 'participants') zijn er allemaal ? In elk geval horen hier ook bij : router, remote clients (peers), nameserver, enz. (Entiteiten op het niveau van klassen worden niet gevraagd). Bespreek je sequence diagram met de practicumdocent.

Maak een state diagram van de volledige XChange PC component.

Teken een deployment diagram van de Xchange applicatie zoals die uiteindelijk gaat werken. Denk ook hier weer aan alle verschillende verschillende systemen.

**OPDRACHT 2: [15%] ONTWERP VAN DE XCHANGE ROUTER COMPONENT**

Specificeer, uitgaande van het sequence diagram, alle interfaces (berichten/protocollen) van de router component (voor zover nog niet beschreven in onderstaande appendix). Beschrijf ook het ontwerp van de router functies.

**OPDRACHT 3: [10%] TESTRAPPORT**

Maak een testplan voor de volledige XChange applicatie waarin je beschrijft hoe je het testen gaat aanpakken. Hierin ook een plaatje van de testconfiguratie (welke verbindingen en welk proces op welke machine). Beschrijf de testcases. Hierbij kun je uitgaan van de requirements die je in week 3 hebt opgesteld en de globaal geschetste use cases (beschikbare menu opties). De resultaten van de testuitvoering moeten worden vastgelegd in een testrapport.

#### OPDRACHT 4: [50%] IMPLEMENTATIE

Implementeer de router component in C. Pas ook de Java code van de PC component aan om met de router te kunnen communiceren.

NB 1: Twee handige links m.b.t. sockets en threads zijn [Sockets](#) en [Threads](#)

NB 2: Als je gaat compileren met threading, zorg er dan voor dat de compiler de thread library kan vinden via de optie "-lpthread".

Belangrijk is zinvol commentaar. In elk geval voor elke functie aangeven wat deze functie doet, wat is de preconditionie (toestand voor aanroep) en postconditie (resultaat na afloop). En uitleg bij de belangrijkste datastructuren.

Je kunt alles onder Linux op een desktop/laptop ontwikkelen. Als het programma volledig werkt hoef je alleen nog te crosscompileren met de crosscompiler (mipsel-linux-gcc) en het resultaat te kopiëren naar de router via scp.

Schrijf ook een eenvoudige handleiding voor gebruiker waarin je iets vertelt over :

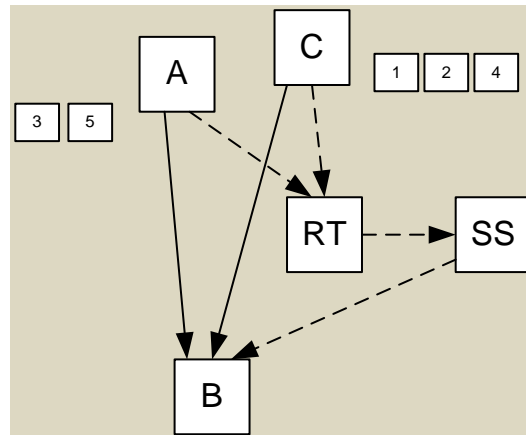
- hoe installeren
- welke bestanden horen bij de XChange applicatie zelf en hoe zijn die opgebouwd
- hoe gebruik je de applicatie
- wat werkt niet / wat zit er niet in

Een demonstratie van de XChange router telt mee voor het cijfer. Laat bij de demonstratie o.a. zien dat de router het downloaden over kan nemen en dat een peer tijdens een download tijdelijk offline kan gaan. Bij de demonstratie moet de router applicatie (uiteraard) op de router draaien en moeten de peers op *verschillende machines* draaien (dus geen localhost demo).

**OPDRACHT 5: [BONUS] VAN NAMESERVER NAAR TRACKER**

In de huidige strategie worden blokken achtereenvolgens binnengehaald van een peer (voor zover ze beschikbaar zijn). Een veel betere strategie is wanneer tegelijk verschillende blokken van verschillende peers (in willekeurige volgorde) kunnen worden binnengehaald.

Stel er zijn 3 peers A, B en C. Peer A wil een file downloaden die uit 5 blokken bestaat. Stel B heeft blokken 3 en 5, en C blokken 1, 2 en 4. Dan zou A deze tegelijkertijd (in 2 threads) kunnen binnenhalen.



Bij een implementatie van deze strategie zijn in elk geval de volgende uitbreidingen nodig :

- “upgraden” van de nameserver : deze moet – net als een tracker – gaan bijhouden welke peer welke blokken heeft
- aanpassing van het protocol, zodat de tracker aan een peer kan vragen : “welke *blokken* heb jij beschikbaar ?”
- aanpassen van de client GUI (multi-selection in de SearchFileFrame )
- multi-threading aspecten : het downloaden van een enkele file verdelen over meerdere threads
- hoe ga je dan om met het overnemen van het downloaden door de router ?

## APPENDIX – PROTOCOL BESCHRIJVING

### NAMESERVER

Alle berichten worden afgesloten door “\n” oftewel end-of-line.

<sp> is een spatie.

#### 1. Opvragen lijst van peers

Client: LIST

Server: <IP1><sp><IP2><sp>. . . <sp><IPn>

Parameters : <IPi> is het IP adres van een peer i.

De server antwoordt met een lijst van aangemelde clients gescheiden door spaties.

Server: FAIL

De server antwoordt met een FAIL als er een fout is opgetreden.

#### 2. Toevoegen client

Client: ADD " "<IP>

Parameters: <IP> is het adres van de aan te melden client.

Server: OK/FAIL

#### 3. Verwijderen client

Client: REMOVE " "<IP>

Parameters: <IP> is het adres van de aan af te melden client.

Server: OK/FAIL



## ALGEMEEN

Alle (tekst)berichten worden afgesloten door “\n” oftewel end-of-line.

<sp> is een spatie.

Uitgangspunt is dat filenamen geen spaties bevatten.

## STORAGESERVER

### 1. Plaatsen van een block

Client: POST<sp> <filename><sp><filesize><sp><blocknr><sp><blocksize><sp><blockdata>

Parameters:

- <filename> is de naam van de het te uploaden bestand
- <filesize> is de grootte van de file in bytes
- <blocknr> is het rangnummer van het blok, te tellen vanaf 0
- <blocksize> is de grootte van het blok
- <blockdata> is de inhoud van het blok

Server: OK/FAIL

### 2. Halen van een block

Client: GET<sp> <filename><sp> <blocknr>

Parameters:

- als bij POST

Server: OK byte[0] byte[1] . . . byte[n]

Server: FAIL

De server antwoordt met een FAIL als het block niet succesvol is weggeschreven doordat de file niet bestaat, een incorrecte grootte heeft of de blocknr niet klopt of niet bestaat.

### 3. Verwijderen van alle bestanden

Client: REMOVEALL

Parameters:

Server: OK/FAIL

## XCHANGE PEER (CLIENT)

### 1. Halen van een block van peer

Client: GET<sp><filename><sp><blocknr>

Parameters:

- <filename> is de naam van de te up- en downloaden bestand.
- <blocknr> is het rangnummer van het block, het eerste block heeft nummer 0.

Server: OK byte[0] byte[1] . . . byte[n]

Server: FAIL

De server antwoordt met een FAIL als de file niet bestaat of het blocknr niet klopt.

### 2. Zoeken naar een bestand

Client: SEARCH<sp> pattern

Parameters:

- <pattern> is de pattern van de naam. Alleen de volgende twee patterns worden ondersteund:
  - \* : geef alle bestaande bestanden;
  - <pattern> : geef alle bestanden waarbij <pattern> als substring in de naam voorkomt.

Server: OK<sp><filename[1]><sp><filesize[1]><sp> . . . <filename[n]><sp><filesize[n]><sp>

met :

- <filename[i]> is de i-de filename die voldoet aan <pattern>;
- <filesize[i]> is de grootte van file[i].

Server: FAIL

De server antwoordt met een FAIL als de aanvraag niet klopt of een andere fout optreedt.

## XCHANGE PC → XCHANGE ROUTER PROTOCOL

Je dient dit protocol zelf te ontwerpen. Het protocol zal in elk geval in het starten en stoppen van de router download moeten ondersteunen. (Ook moet de storageserver nog verder worden geïmplementeerd).