# SystemY

## Bestand

Klasse om een bestand in te bewaren

### CalculateHash() Deze functie berekent de hash voor deze node zelf bij initialisatie.

### getHash() Zal de hash teruggeven.

### getNaam() Zal de naam teruggeven.

### getFile() Zal het bestand teruggeven.

### getPath() Zal het pad teruggeven.

### changeName(String nieuweNaam) Methode voor het veranderen van de bestandsnaam, de hash wordt als parameter teruggegeven.

### checkName(String checkNaam) Checken of de hash effectief bestaat.

## BestandenLijst

Klasse voor een lijst van bestanden van een node in te bewaren.

### addBestand(String naamBestand, String pathBestand)

### Voegt een bestand toe aan de bestandenlijst.

### getBestand(String naamBestand)

### Zal bestand teruggeven aan de hand van de bestandsnaam.

### verwijderBestand(Bestand teVerwijderen) Zal een bestand verwijderen uit de lijst.

### verwijderBestandMetNaam(String naamBestand) Zal een bestand verwijderen uit de lijst a.d.h.v. zijn bestandnaam.

## MulticastReceiverThread

Thread om multicast pakketten te ontvangen.

### Run() Zal een datagram pakket aanmaken om te ontvangen.

### Close() Multicast groep verlaten.

## NamingServer

Houdt enkel een lijst van nodes bij met hierin de naam, de hash en het IP adres.

### getFileLocation(String fileName) Zal teruggeven op welke node de file is gelokaliseerd.

## Nodelijst

Zal een lijst bijhouden van de verschillende nodes.

### addNode(String nameIp)

### Voegt een node toe aan de TreeMap.

### removeNode(int place)

### Verwijdert de node terug.

### calculateHash(String nodeNaam) Berekent dezelfde hash als in de node functie, afhankelijk van de naam van de node.

## NodeNamingServer

Alle unieke nodes zoals ze door de namingserver worden bijgehouden.

### setName(String name)

### Naam van de node instellen.

### getName()

### Geeft de naam van de node terug.

### setIpAdress(String ipAdress)

### Ip adres van de node instellen.

### getIpAdress()

### Geeft het Ip adres terug van de node.

### calculateHash()

### Berekent de hash van de naam.

### compareHash(String nodeNaam)

### Als beide nodenamen dezelfde hash uitkomen, wordt “true” teruggegeven.

### getHash()

### Zal de hash teruggeven.

## RMIServer

Dit is de RMI server.

# SystemY Client

## MulticastReceiverThreadClient

Multicast berichten ontvangen met de client.

### run()

### Datagram pakket aanmaken en ontvangen.

### close() Multicast groep verlaten.

### calculateHash(String nodeNaam)

### Berekent de hash op basis van de naam van de node.

## MulticastSender

## Verzenden van een multicast (door de node bij het opstarten).

### readConsoleIP()

### Zelfde maar dan wel voor het IP adres. (Kunnen we eigenlijk ook ophalen, dus wordt eigenlijk niet gebruikt.)

### sending(String naam, String Ip)

### We verzenden de naam en de het IP-adres van de node die in het systeem komt.

### close() We verlaten de multicast groep.

### calculateHash(String nodeNaam)

### Zal de hash berekenen op basis van de nodenaam.

## NodeClient

Dit is de effectieve node. Deze bevat een nodelijst van de andere nodes in het systeem. Deze bevat ook een bestandlijst met alle bestanden die zich op de node bevindt. **(Bevat ook een replica)**

### ConsoleGUI()

### Zal de gevraagde methode uitvoeren.

### checkLocalFiles(File dir)

### Zal de lokale files oplijsten. (Bestandlijst afdrukken)

### getFileLocation(String fileName) Zal de locatie van een bepaalde file teruggeven.

### startUp() RMI verbinden met de naming server.

### addNode(int hash, String ipadres) Voegt een node toe aan de nodelijst van de respectievelijke node.

### calculateHash(String nodeNaam) Berekent de hash op basis van de nodenaam.

### readConsole()

Zal de input vanuit de console teruggeven. (Is nodig in de consoleGUI()-methode om een keuze te maken.)

### shutdown() Zal voorgaande node’s next node aanpassen naar de volgende node ipv deze node. En zal de volgende node’s previous node aanpassen naar de vorige node ipv deze node.

### notifyNext(int ownHash, int nextNodeHash, int Hash) Zal via RMI de volgende node informeren. 🡪 Deze methode kan gebruikt worden bij zowel shutdown als bij de startup van een node.

### notifyPrevious(int previousNodeHash, int ownHash, int Hash) Zal via RMI de vorige node informeren. 🡪 Deze methode kan gebruikt worden bij zowel shutdown als bij de startup van een node

### getNotified(int previousHash, int nextHash) De node moet ook geinformeerd kunnen worden bij veranderingen int het system. Ook via RMI.

### readConsoleName() Zal de naam van de node teruggeven. Zodat je deze bij initialisatie de node een naam kan geven. (Wordt uit de console uitgelezen.)

### 

### 

### 