# Computer Science 2B

Practical Assignment 05

2017-08-28

Deadline: 2017-09-19 12h00 Marks: 50

This practical assignment must be uploaded to eve.uj.ac.za **before** 2017-09-19 12h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

The JDK has been installed on the laboratory computers along with the Eclipse IDE.

## This practical will focus on peer-to-peer file sharing with UDP.

One of the most popular applications that use peer-to-peer communication is peer-to-peer file sharing (where examples are BitTorrent and Direct Connect). Instead of downloading a file from one host or server, it is downloaded from a peer that has the file. The file is seeded (or sent) from a peer to leechers that want the file (or a peer that does not have the complete file).

In this practical, we will be creating a UDP-based peer-to-peer file sharing system which consists of a Client that can either send files (Seeder mode) or receive files (Leecher mode). Once a client starts, it asks the user for which mode the client will be using and the user can commence sending or receiving binary files depending on what mode it is in, until the user terminates the connection by closing the client. In order to achieve this, you require a client that has two modes:

### Seeder Mode

In Seeder mode the following functionality needs to be provided:

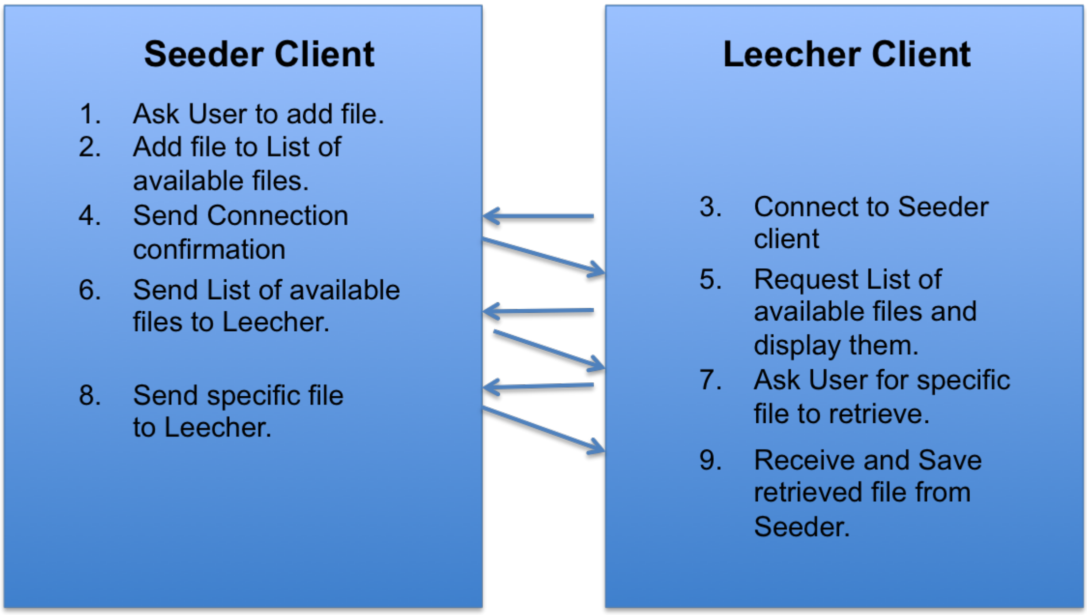
1. The Seeder must provide the user with the ability to add a file to the list of available files that can be shared. **Suggestion** - You can use an add button that chooses files with a *JFileChooser* and a *JList* to display added files.
2. It must be able to send a **list of files** available (it can be a numbered list) to another peer that has connected to it.
3. It must be able to send the **chosen local file** to the peer that asked for it.

## Leecher Mode

In Leecher mode the following functionality needs to be provided:

1. The Leecher must connect to a Seeder client (using a specific host address and port number). **Suggestion** - You can use 2 *JTextFields* to capture the host address and port number from the user, along with a Connect button.
2. Ask the Seeder peer what files it has available and provide the user with the ability to choose a specific file. **Suggestion** - A *JList* can be used to display the available files from a Seeder client.
3. Tell the Seeder peer, which file it wants (you can send the number, the user chose). **Suggestion** - A Retrieve button can be used to send the selected index to the Seeder client.
4. Receive and save the file from the Seeder peer.

## Example



**Seeder Client**

1. Ask User to add file.
2. Add file to List of available files.
4. Send Connection confirmation
6. Send List of available files to Leecher.
8. Send specific file to Leecher.

**Leecher Client**

3. Connect to Seeder client
5. Request List of available files and display them.
7. Ask User for specific file to retrieve.
9. Receive and Save retrieved file from Seeder.

# Bonus

1. Provide the ability to send multiple files at once.

# Marksheet

| | |
|---|---|
| 1. Client GUI for both Seeder and Leecher modes. | [5] |
| 2. Seeder Mode: Send list of files available. | [5] |
| 3. Seeder Mode: Send the chosen local file. | [10] |
| 4. Leecher Mode: Ask peer what files it has available. | [5] |
| 5. Leecher Mode: Receive a file from a remote peer. | [10] |
| 6. Correct execution | [15] |
| 7. Send multiple binary files at once. | [10 (bonus)] |