

CSCE441101 – Fundamentals of Distributed Systems

Fall 2020 Term Project: Remote Fork Implementation

Supervised by Prof. Amr El-Kadi

Feras Awaga

900162227

Design:

To be able to implement the myfork() function which sends a process from one machine to another machine then continue running from the same position in which the myfork() function was called on the sending machine.

The first step to do so was to use socket programming to be able to send files between machines. Attached in the files are two programs. A client program and a server program. Those two programs connect to each other through the use of sockets and IP address of each other to be able to send files across different hosts.

The second step was to be able to create a checkpoint at a certain point in a running process to be able to restart it on another machine. To do so, I used a package that can be install in Ubuntu called DMTCP. This uses commands in the terminal to start a process, create a checkpoint and restart from where the process was checkpointed.

Steps to implement project:

1. Install VirtualBox
2. Create two different virtual machines both having the latest Ubuntu distribution.
3. Open the VirtualBox preferences to create a NAT network so that both virtual machines can communicate with each other. To do so,
   1. Open preferences
   2. Network
   3. Add new
   4. Choose a name
   5. Press Ok
   6. Go to the settings of the first virtual machine
   7. Go to Network
   8. In the “Attached to” options, choose “NAT Network”
   9. Choose in the “Name” options the name you created in step d.
   10. Repeat from steps f to i for the second virtual machine.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

1. To download and run DMTCP on your device, visit the attached link and download the latest version.

<https://sourceforge.net/projects/dmtcp/files/>

1. Extract the file to the desktop.
2. Open the terminal from the location in which you extracted the file
3. Type in : ./configure
4. Type in: make
5. Type in: sudo make install
6. Open Vscode on both machines.
7. On the first machine, add to one folder the three cpp files in the Machine 1 folder attached.
8. On the second machine, add to one folder the three cpp files in the Machine 2 folder attached.
9. To send from Machine 1 to Machine 2:
   1. Open the terminal in folder of the second machine.
   2. Type: g++ server.cpp -o server
   3. Type: ./server
   4. Now the server is running and waiting for the client
   5. On the first machine, open the terminal in folder:
   6. Type: g++ client.cpp -o client
   7. Type: g++ test.cpp -o test
   8. Type: dmtcp\_launch ./test
10. To send from Machine 2 to Machine 1:
    1. Open the terminal in folder of the second machine.
    2. Type: g++ server2.cpp -o server2
    3. Type: ./server2
    4. Now the server is running and waiting for the client
    5. On the first machine, open the terminal in folder:
    6. Type: g++ client2.cpp -o client2
    7. Type: g++ test2.cpp -o test2
    8. Type: dmtcp\_launch ./test2

Dependencies:

1. Both virtual machines must be on the same network.
2. DMTCP library must be installed on both virtual machines.
3. VsCode must be installed and configured to run cpp programs on both virtual machines.

How to know the project is working:

I printed on each device that is forked from “This is the parent” 10000 times and on the other machine “This is the child” a thousand times. You first must run the server then launch the program that calls myfork and everything is done automatically (sending and restarting).