Stage 1: HTTP daemon on Renesas M16 board

Cuong Phu Nguyen Kyung Chul Lee

URL: http://www4.ncsu.edu/~kclee/csc714

1. Summary

This report covers the time period October 18, 2005 – November 1, 2005 and encompasses Stage 1 - "Get familiar with the Renesas board and environment" and a part of Stage 2 - "Design a RAM file system and a set of API" of the project plan. It aims both to describe progress made and discuss the issues encountered.

The achievements of our group so far include:

- Studied and experienced with M16 environment from running the given HTTP daemon code.
- Designed and specifying rules and algorithms for our file system and currently testing the design on the Linux system first.

2. Plan

We had the original plan as following for the project:

- Stage 1 (October 18 – 29)
  Get familiar with the Renesas board and environment – Examine and run provided same codes and learn the development environment.

- Stage 2 (October 30 – November 14)
  Design a RAM file system and a set of API – Come up with a feasible file system for a M16 board and create specifications.

- Stage 3 (November 15 – November 22)
  Improvement for the HTTP daemon - Find inefficiency and possible bottlenecks, and improve over those findings.

- Stage 4 (April 23 – April 28)
  Test module - Create possible scenarios and run multiple tests based on the scenarios.

3. Accomplishments & Issues

We have completed Stage 1 and a part of Stage 2.

Stage 1

From the CD we have received from Dr. Mueller, we installed all the necessary programs onto our computer. At this time we needed the serial number of the M16 board in order to continue the installation of the program. We had to detach the M16 real board from the Ethernet board because the serial number was written on the back of the M16 real board. After then, it went pretty smoothly and we were able to start compile and upload the programs to the M16.

Stage 2

We have studied various file systems such as Simple File System, Transaction-Safe FAT File System, RAM File System, etc. As of beginning, we decided enforce the following restriction in order to simply the process.

- The name of files and directories can be maximum 10 letters.
- The number of files and directories can be maximum 20.
- Creating a file is allowed but deleting is not permitted.
- Open for read and write are allowed but append is not permitted.
- In order to create a new file, one must specify the size of the file.
- We provide a file level lock protection. We allow only one writer at any time.

These restrictions are remained for the future extension and can be improved over time.

We are currently implementing this file system on the Linux environment first in order to prove the concept of the algorithm.

4. Conclusion and Next Step

Our next step is to complete the implementation on Linux and move to M16. Once we are able to run the HTTP daemon with our file system, we are going to find inefficiency in the code and try to improve possible bottlenecks.