Kernel 2
CS452 - Spring 2014
Real-Time Programming

Team
Max Chen - mqchen
mqchen@uwaterloo.ca

Ford Peprah - hkpeprah
ford.peprah@uwaterloo.ca

Bill Cowan
University of Waterloo
Due Date: Friday, 30^{th}, May, 2014

Table of Contents

1 Program Description

1.1 Getting the Program

To run the program, one must have read/write access to the source code, as well as the ability to make and run the program. Before attempting to run the pogram ensure that the following three conditions are met:

- You are currently logged in as one of `cs452`, `mqchen`, or `hkpeprah`.
- You have a directory in which to store the source code, e.g. `~/cs452_microkern_mqchen_hkpeprah`.
- You have a folder on the FTP srever with your username, e.g. `/u/cs452/tftp/ARM/cs452`.

First, you must get a copy of the code. To to this, log into one of the aforementioned accounts and change directories to the directory you created above (using `cd`), then run one of

```
git clone file:////u8/hkpeprah/cs452-microkern -b kernel2 .
                        or
git clone file:////u7/mqchen/cs452/cs452-microkern -b kernel2 .
```

You will now have a working instance of our kernel2 source code in your current directory. To make the application and upload it to the FTP server at the location listed above (`/u/cs452/tftp/ARM/YOUR_USERNAME`), run `make upload`.

1.2 Running the Program

To run the application, you need to load it into the RedBoot terminal. Ensure you've followed the steps listed above in the "Getting the Program" settings to ensure you have the correct directories and account set up. Navigate to the directory in which you cloned the source code and run `make upload`. The uploaded code should now be located at

`/u/cs452/tftp/ARM/YOUR_USERNAME/assn2.elf`

To run the application, go to the RedBoot terminal and run the command

```
load -b 0x00218000 -h 10.15.167.4 ‘‘ARM/YOUR_USERNAME/assn2.elf’’; go
```

The application should now begin by running through the game tasks before reaching a prompt. The generated files will be located in `DIR/build` where `DIR` is the directory you created in the earlier steps. To access and download an existing version of the code, those can be found at `/u/cs452/tftp/ARM/mqchen/assn2.elf` and `/u/cs452/tftp/ARM/hkpeprah/assn2.elf`.

2 Kernel Structure**

2.1 System Calls

3 Game Tasks

3.1 Priorities

Task Name	Task ID	Priority
FirstTask	0	15
NameServer	1	15
Server	2	11
Client (OCDDX)	3	6
Client (BITTG)	4	7
Client (RNFKS)	5	4
Client (YCWTD)	6	0
Client (FWGJH)	7	6
Client (UQSTV)	8	2
Client (YCJLC)	9	0
Client (HGMSK)	10	5
Client (MSEYY)	11	7
Client (GEPMY)	12	5

3.2 Game Task Output

The output from the GameTask is as follows:

```
Player BITTG(Task 4) throwing PAPER
Player MSEYY(Task 11) throwing PAPER
Round was a TIE
Press any key to continue:

Player BITTG(Task 4) throwing PAPER
Player MSEYY(Task 11) throwing PAPER
Round was a TIE
Press any key to continue:

Player BITTG(Task 4) throwing ROCK
Player MSEYY(Task 11) throwing PAPER
MSEYY won the round
Press any key to continue:
```

```
Player OCDDX(Task 3) throwing ROCK
Player FWGJH(Task 7) throwing PAPER
FWGJH won the round
Press any key to continue:

Player HGMSK(Task 10) throwing ROCK
Player GEPMY(Task 12) throwing ROCK
Round was a TIE
Press any key to continue:

Player HGMSK(Task 10) throwing PAPER
Player GEPMY(Task 12) throwing ROCK
HGMSK won the round
Press any key to continue:

Player RNFKS(Task 5) throwing PAPER
Player UQSTV(Task 8) throwing ROCK
RNFKS won the round
Press any key to continue:

Player YCWTD(Task 6) throwing ROCK
Player YCJLC(Task 9) throwing PAPER
YCJLC won the round
Press any key to continue:
```

The implementation of `random` using a set seed, so the results from the game are deterministic, which allows us to argue that the results will always be the same as above. First, the explanation of how Rock-Papers-Scissors works. To begin a game of Rock-Paper-Scissors, two parties must agree to play, at which point, each party throws one of {Rock, Paper, Scissors} simulataenously. Rock beats Scissors, Scissors beats Paper, and for some god awful reason, Paper beats Rock. If both parties throw the same hand, the round ends in a tie, and neither party is victorious.

4 Performance Measurements

4.1 Results

Message Length	Caches	Send Before Receive*	Optimization	Microseconds
4 bytes	off	yes	off	343.8453713
64 bytes	off	yes	off	462.8687691
4 bytes	on	yes	off	24.41505595
64 bytes	on	yes	off	31.53611394
4 bytes	off	no	off	378.4333672
64 bytes	off	no	off	496.439471
4 bytes	on	no	off	27.46693795
64 bytes	on	no	off	35.60528993
4 bytes	off	yes	on	192.2685656
64 bytes	off	yes	on	231.9430315
4 bytes	on	yes	on	12.20752798
64 bytes	on	yes	on	15.25940997
4 bytes	off	no	on	215.6663276
64 bytes	off	no	on	255.3407935
4 bytes	on	no	on	14.24211597
64 bytes	on	no	on	16.27670397

***** - Assignment says "Send Before Reply", however, replies are non-blocking and don't depend on a send to occur.

4.2 Explanation

Something something...

5 MD5 Hashes

Source files can be accessed at either `/u7/mqchen/cs452/cs452-microkern` or `/u8/hkpeprah/cs452-microkern`. The MD5 hashes of the source files are as follows: