CIS552: Advanced Programming

Handout 16

Finite Channels

My Solution (with Peng Li)

```
data FiniteChan a = FiniteChan {
    readCell  :: MVar (Stream a)
  , writeCell :: MVar (Stream a)
  }
type Stream a = [MVar a]

newFiniteChan :: Int -> IO (FiniteChan a)
newFiniteChan size = do
  q <- replicateM size newEmptyMVar
  r <- newMVar (cycle q)
  w <- newMVar (cycle q)
  return (FiniteChan r w)

-- where    cycle l  =  l ++ l ++ l ++ ...
```

My Solution (with Peng Li)

```
data FiniteChan a = FiniteChan {
    readCell  :: MVar (Stream a)
  , writeCell :: MVar (Stream a)
  }
type Stream a = [MVar a]

readFiniteChan :: FiniteChan a -> IO a
readFiniteChan (FiniteChan r w) = do
  (hd:tl) <- takeMVar r
  x <- takeMVar hd
  putMVar r tl
  return x
```

My Solution (with Peng Li)

```
data FiniteChan a = FiniteChan {
    readCell  :: MVar (Stream a)
  , writeCell :: MVar (Stream a)
  }
type Stream a = [MVar a]

writeFiniteChan :: FiniteChan a -> a -> IO ()
writeFiniteChan (FiniteChan r w) x = do
  (hd:tl) <- takeMVar w
  putMVar hd x
  putMVar w tl
```

Final Projects

Schedule

Apr 9	STM
Apr 14	Composing Financial Contracts (guest lecture)
Apr 16	Generalized Algebraic Datatypes (guest lecture)
Apr 21	Final project presentations
Apr 23	Final project presentations
Apr 28	Final project presentations

Final Project Milestones

Apr 9	STM
Apr 14 Apr 16	Composing Financial Contracts (guest lecture) Generalized Algebraic Datatypes (guest lecture) Initial design document (3-page document)
Apr 21 Apr 23	Final project presentations Final project presentations Working prototype of core functionality (screenshots)
Apr 28	Final project presentations Code reviews (*)
When? (**)	Final version

(**) Do people want to do this?

(**) We need to choose a date between April 30 and May 5

Project Logistics

- Groups of 2 (probably with one group of 3)
- Work with any partner(s) you want—it's fine if you've already worked together on a project this semester
- Use any language or combination of languages, so long as the bulk of the code is written in some functional language (Haskell, OCaml, Scheme, etc.)
- Aim to spend approximately 30 hours total

Project Topics

- Carte blanche: Choose any programming problem that interests you

Default Topic

- I think it would be fun if several groups ended up working on variants of the same idea.
- I propose this one:

  A collaborative virtual environment

  (i.e., a better Second Life / Sims / etc.)
- Building a real one of these is a gigantic task—need to attack just a piece of the problem...
  - 3-d modeling
  - distributed simulation
  - a new "spatial scripting language"
  - etc., etc.