21W.730
April 30, 2001

Multi-Threaded Thing ™

## ASM

```
     mov  ax, 0200h
     mov  dl, 66
     inc  cx                  ; Prints ‘C’
     int  21h                 ; (My first initial)
     mov  CEC, Chris
     mov  cx, 180             ; time, in minutes
LearnAssembler:
     call writeProgram
     push Nathan
     push Breiland
     push Curtis              ; set up to
     push Gordy               ; takeBreak with Friends
     call takeBreak
     loop LearnAssembler

     mov  Bus, Chris
     mov  Home, Bus
     mov  House, Chris
     call makeSnack
     mov  LivingRoom, Chris
     call eatSnack
     call watchTelevision
     iret                     ; exit this subroutine
                              ; (take a nap)
```

## C++

```
class Student {
        float fGpa;
        int iClass, iMajor;
        boolean bDoubleMajor;
};

void main( ) {
        Student *Chris = new Student( );
        Chris→applyForInternship(SandiaNationalLabs);
        SandiaNationalLabs→acceptStudent(Chris);
        RoboticVehicleRange→hireStudent(Chris);
        Chris→learn(BorlandCBuilder);            // learn visual C for the RVR
}
```

## Scheme (lisp)

```
(apply Chris MIT)
(accept MIT Chris)
(set! Chris 'MIT-Student)
(rush Chris)
(stayInTempDormRoom Chris)
(getBored Chris)
(visit Chris EpsilonTheta)
(playGames Chris EpsilonTheta)
(define ET EpsilonTheta)
(pledge Chris ET)
(learn Chris 'Scheme)      ;; learn Scheme for 6.001
```

## Java

```
/** Main is the main entry point to this program.
 *  There are no defined command-line arguments
 *  As of April 30, 2001:
 *       Creates new Student, moves that student to the East Campus dorm at MIT
 *       Has student work a summer internship at Sandia National Labs
 *       Makes student learn Java
 **/
        public class Main extends Object {

                public static void main(String args[] ) {
                        Student Chris = new Student( ‘03, 6-3 );
                        Chris.moveTo(MIT.EASTCAMPUS);
                        Chris.summerInternship(DOE.SandiaNationalLabs);
                        Chris.returnTo(MIT, “Fall 01”);
                        Chris.learn(“java”);
                }
        }
```

## Asm

```
call EatDinner
cmp  dueTomorrow, wantToDoHomework
jna  takeNap         ; See if importance of homework is
                     ; greater than my desire to skip
xor  bx,bx
mov  cx, 2000h
mov  ah, 39h
```

```
     ld   dx, MathBook
     int  21h                ; Find mathbook, do homework
takeNap:
     mov  Bed,Chris
     mov  cx,0FFFFh
sleep:
     pushf                   ; push flags
     pop  ax
     and  ax,20h             ; In case something wakes me up
     jnz  wakeUp
     loop sleep              ; Keep sleeping until something
                             ; wakes me up.
```

## C++

```
Time GO_HOME_TIME = new Time( 5:00 PM );

void main( ) {
        Chris→leaveWorkAt(GO_HOME_TIME);
        Chris→driveHome( );

        if (dinner→looksInteresting( ) )
                Chris→eatDinnerWithFamily( );
        // Dinner has a low probability of looking interesting ( < 0.25)

        else {
                // If dinner does not look good, flip a coin to see whether it is
                // Microwave Pizza or Microwave Hamburger tonight.
                if (random( ) > 0.5)
                        Chris→makeMicrowavePizza( );
                else
                        Chris→makeMicrowaveHamburger( );
                Chris→eatDinnerAlone( );
        }
        Chris→goToRoom( );
        turnOnTV( true );
        turnOnRadio( true );
        turnOnComputer( true );
        if (Chris→haveGoodBook( ) )
                Chris→read( );
        else {
                if ( Chris→tired( ) )
                        break;          //exit this part (continue on to sleep)
                else if ( Breiland→isHome( ) ) {
                        Chris→inviteFriend( Breiland );
                        playComputerGame( Chris, Breiland );
                }
                else
```

```
            Chris→programComputerGame( );
    }
    while ( nothingHappens( ) )
        Chris→sleep( );
}
```

## Scheme (lisp)

```
(if (> (ask Chris ’laziness) (ask Chris ’desire-to-go-home))
  (begin
    (go Chris ’APO-office)
    (wait Chris ’van-ride-home))
  (walk Chris ’home))
                ;; Walk home takes 30 minutes, sometimes
                ;; didn’t feel like walking
(sleep Chris)
(if (ask Chris ‘does-dinner-look-good)
  (eat Chris)
  (begin
    (make-ramen)
    (eat Chris)))

(if (> (length
    (filter (lambda (person) (ask person ’want-to-play-board-game))
        (ask ET ’people-at-home))) 0)
  (play-board-game Chris)
  (if (ask Chris ’have-homework-due-tomorrow)
    (do-homework Chris)
    (if (ask Chris ’have-good-book)
      (read Chris)
      (play-computer-game))))
                ;; see if anyone wants to play, if not, see if I
                ;; have homework, if not, either read or play
```

## Java

```
/** Main is the main entry point to this program.
 *  There are no defined command-line arguments
 *  As of April 30, 2001:
 *       Creates second Student, Shelley.  Creates couple containing Chris and Shelley
*       Has couple flip coin to decide either Pizza or Spaghetti for dinner
*       Couple watches a Simpsons episode selected from the episodes available
*       Chris writes a paper, while Shelley does homework or work for her UROP
**/
        public class Main extends Object {
                Food food1 = new Food(“Spaghetti”);
                Food food2 = new Food(“Pizza”);
                Student Shelley = new Student(’03, 6-2);

                public static void main(String args[] ) {
                        Couple us = new Couple(Chris, Shelley);
                        us.makeDinner((Math.random( ) > 0.5) ? food1 : food2);
                        us.eatDinner( );
                        us.watchEpisode( selectSimpsonsEpisode ( ));
                        us.takeNap( 2:00 );
                        Chris.writePaper( );
                        if (Math.random( ) > 0.5)
                                Shelley.doHomework( );
                        else
                                Shelley.doUropWork( );
                }
        }
```

This paper is an autobiography. It explains comparable aspects of four different periods in my life: Sophomore/Junior year in high school (in assembly); Senior year/summer after graduation (in C++); First ¾ of freshman year at MIT (in Scheme); and the time since then (in Java). The first assembly section describes an average day at CEC, the Career Enrichment Center, where I spent 3 hours a day during my Sophomore and Junior years in high school. The first C section is about the application process and beginning of time spent at an internship at Sandia National Labs, a national research facility with funding from the Department Of Energy. The first Scheme section does the same for my applying to MIT. The first Java section mentions a few of the changes I underwent in moving out of the Independent Living Group and into a Dormitory with my girlfriend. The second section for all of these languages describes an average day of life at home after classes or work was finished.

Some notes and observations:

Java and C are more disassociated languages, which seem to coincide with times in my life when I was less socially active. Scheme on the other hand corresponds to a time when I was the most socially active I've been and also a time when my schedule was less regular. Scheme basically just uses one big heap where everything's located, and there are no rules about accessing particular variables; everything is accessible and easy to change (less rigid).

Courtesy of Anonymous Student. Used with permission.