

From: John Sarraillé <john@ishi.csustan.edu>
Subject: **Re: Solution for foodPass**
Date: October 28, 2010 13:17:34 PDT
To: undisclosed-recipients::;



From: John Sarraillé <john@ishi.csustan.edu>
Date: October 28, 2010 13:15:43 PDT
To: Julie D Gorman <julie@cs.csustan.edu>
Subject: Re: Solution for foodPass

Thanks Julie --

It looks like the 'foodPass' code is due for some re-writing to conform to changes in C++ specifications. For the time-being, I'll modify the Makefile for the Mac computer that I posted to include the -fpermissive flag and I'll pass along advice to the students to 'ignore the warnings' they get when compiling on the Macs. I tried this on my Mac, and the compiled code seemed to pass a few preliminary tests without error. As a temporary precaution, I'd also advise students to include testing the code on one of the sun workstations as a part of their process of developing a solution to the problem.

-- js

```
=====
John Sarraillé, Professor, Comp. Sci.
CSU Stanislaus || john@ishi.csustan.edu
(209) 667-3345 || (209) 634-1904
=====
Unity
=====
```

On Oct 28, 2010, at 10:16, Julie D Gorman wrote:

```
John,

If we add

-fpermissive

to the g++ statements in the Makefile, foodPass will compile.

#
# Note: you need to make sure that the indented lines below
# are made with TABS. If you just cut and paste from your
# browser window, probably the tabs will be 'translated' into
# a series of blanks. So check and change the indents back
# into tabs or you will get syntax errors.
#

foodPass: foodPass.o sem.o
    g++ -fpermissive -o foodPass foodPass.o sem.o
foodPass.o: foodPass.cpp sem.h
    g++ -fpermissive -c foodPass.cpp
sem.o: sem.cpp sem.h
    g++ -fpermissive -c sem.cpp

This may cause other errors to also be ignored but warnings will be issued.

julie@goldberry.csustan.edu:(~/programming/os) make
g++ -fpermissive -c foodPass.cpp
foodPass.cpp: In function 'void* Diner(void*)':
foodPass.cpp:253: warning: cast from 'void*' to 'int' loses precision
g++ -fpermissive -c sem.cpp
sem.cpp: In function 'void enq_sem_Q(sim_sem_data*, sim_PCB*)':
sem.cpp:97: warning: cast from '_opaque_pthread_t*' to 'int' loses precision
sem.cpp: In function 'void serve_sem_Q(sim_sem_data*)':
sem.cpp:118: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:134: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp: In function 'void wait_sem(sim_sem_data*)':
sem.cpp:157: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:172: warning: cast from 'sim_sem_data*' to 'int' loses precision
```

```
sem.cpp:190: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:201: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:219: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp: In function 'void signal_sem(sim_sem_data*)':
sem.cpp:243: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:252: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:264: warning: cast from 'sim_sem_data*' to 'int' loses precision
sem.cpp:276: warning: cast from 'sim_sem_data*' to 'int' loses precision
g++ -fpermissive -o foodPass foodPass.o sem.o
```

From the g++ man page

-fpermissive

Downgrade some diagnostics about nonconformant code from errors to warnings. Thus, using -fpermissive will allow some nonconforming code to compile.

Julie

```
=====
Julie D. Gorman, Computer Science, CSU Stanislaus
One University Circle Turlock, CA 95382 || 209 667-3273
=====
julie@cs.csustan.edu || " All who wander are not lost "
=====
```