CPS 470, Assignment 7, 10 points

Due:

Goal: Simulation of Dijkstra's binary semaphore using a pipe to allow synchronized access to a shared resource under Solaris.

Statement: Implement a binary semaphore using a pipe. A '1' written to the pipe initializes the start value. Read and write operations on the pipe will implement the P and V operations (no wait list is involved as the file system helps in making the operations atomic, a reader is blocked when the pipe is empty, basic property). All operations to pipe using read()/write()/close() calls as needed.

Operations to implement:

int Isem(void): initialize the binary semaphore by writing '1' to pipe,
returns 0 on success and -1 on failure.

int Psem(void): performs the P operation by reading a '1' from the pipe. It
returns -1 on failure and 0 on success.

int Vsem(void): performs the V operation by writing '1' to the pipe. Returns
0 on success and -1 otherwise.

The pipe in question is global to all processes. Stderr is the shared resource among related processes.

Write modular, structured, and well documented C-programs.

Part1: Write a program that will take a count of iterations from command line, create 30 processes (all on the same level in the process tree - think?), each will access the shared resource stderr count times. At each access the process will write the message "My pid: 12345 " to stderr, where 12345 is the pid of the process. Determine the count at which you observe the race condition (interleaved output). Place code in file sharedraw.c

Invocation:

```
./sharedraw <iteration-count>
```

Part2: Write a program that repeats the above experiment but resolves the race condition using the binary semaphore. When run for the same data (iteration-count), it should not produce interleaved output. Place code in file sharedsm.c.

Invocation:

```
./sharedsm <iteration-count>
```

Write a Makefile to produce both executables:

[

```
...
TARG1 = sharedraw
OBJS1 = sharedraw.o
TARG2 = sharedsm
OBJ2  = sharedsm.o

all:    $(TARG1) $(TARG2)

$(TARG1) = $(OBJS1)
        ...

$(TARG2) = $(OBJS2)
        ...
```

]

Turn-in: Submit all relevant files (Makefile and C-source file(s)) via e-mail attached file <global-id>-a7.tar.bz2. Credit by demo only.