CMPSC 472
Fall 2017

Due: November 16, 2017
100 Points

Homework Information

- This assignment can be done in teams of two.
- This assigment is due at the beginning of the class period on the date specified. Late papers are not accepted.
- Please be sure to email a copy of your code using the mail472 command AND hand in a hard copy.
- Name your program using your first initial followed by your last name.
- Add comments at the beginning of your code that include your name(s) and what the program does.
- You should include comments throughout the program and use good variable names. Your comments should explain what your code is doing. You WILL lose points if you do not do this.
- Your hard copy should be "nice" -- no line wraps, no superlarge font. If you have long lines, you should consider using enscript (if you are printing in the SUN lab.) If you print from home, you should consider importing your source code into a word processor so you can control the font size and use the landscape setting. Your hard copy MUST match the files that you email.
- Any programs handed in should compile (and do so in the SUN lab). If they do not compile at all, few, if any points will be awarded. If they do not compile on the Suns (due to long comments, etc.) points will be taken off.
- Please put printouts in a folder with your name and class number on the outside. If you work with someone else, hand in ONE hard copy in ONE folder.
- **Failure to follow these rules WILL result in lost points.**

It's Thanksgiving in BACI-land. We're going to write the necessary code to synchronize some Thanksgiving processes. There are 4 different types of processes we will have to synchronize:

1. Mother process: this process will be responsible for cooking the entire meal. She will prepare the foods for the meal and put them in the oven. When done with this, she must go on a hayride (see the rules for the hayrides below) with the youngest child (as he is afraid to go without his mother). Assume this is child process 8. When done, she enters the house, washes up, puts the food on the table (we will assume the food is done by the time she gets back from the hayride) and then sits down to eat dinner. She must wait to eat until ALL other processes are sitting down and ready to eat and the turkey has been carved. When everyone is done eating and has been excused from the table (done by the Father process), she will go read her favorite book.
2. Father process: this process will hook up the horses to the wagon and begin giving hayrides. (See rules below for giving the hayrides.) When all children and relative processes have been given hayrides, the father process will unhook the horses, feed them, and then go into the house and clean up for dinner. He will then sit down at the table. When everyone is seated, he will carve the turkey. When he has carved the turkey everyone can begin eating. Aftern eating, this process excuses everyone from the table and is done (he goes to take a nap on the sofa and watch the football game).

3. Child process: there will be 8 child processes. The processes start by raking leaves in the yard. When done, each child will line up for a hayride. When done with the hayride, the child will go into the house, clean up, and sit at the table, waiting for everyone else to arrive before eating. NOTE: Child process 8 has a special requirement: the Mother process MUST accompany this child on the hayride; therefore this child must wait on the mother process to be finished preparing the meal. When done eating, the child processes wait to be excused from the table and then are all done (they can go out and play).
4. Relative (aunts and uncles) process: these adult processes (6 total) arrive, go on a hayride, clean up and sit down at the table. As with everyone else, they cannot eat until everyone is seated and the Father process has carved the turkey. They eat, wait to be excused from the table, and then they return home.

The rules are as follows:

1. No one eats until everyone is ready and the turkey has been carved. No one is allowed to leave the table until everyone is done eating. Only the Father process can excuse people from the table.
2. The hayrides are given by the Father process and two horses hooked up to a wagon full of hay. The Father drives the horses around a 100 acre forest and then back to the house. That constitutes one ride. Since the horses are older, the Father process is concerned with them pulling too much weight. Therefore he limits the number of people (in addition to himself) on any one ride to 3 kids OR 1 adult and 2 kids OR 2 adults (this is the order of preference if there is a tie when a choice is to be made). This is a first come, first served situation (after these group rules have been applied). **If you use semaphores, you do not need to worry about the lack of FCFC ordering when removing processes from the queues. If you use monitors, you should enforce FCFS.** The father wants to limit the number of rides, but toward the end, he will give a ride to only one or two children if there is no one else waiting, or to only 1 adult in the same situation.
3. If there are specific situations NOT covered by the above rule, you can opt to deal with them however you choose, but you MUST comment your code to explain what you are doing in that situation.

Given the following: Suppose Child 1 and Child 2 are waiting. Now Relative 1 and Relative 4 arrive. The Father would take Child 1 and 2 along with Relative 1 before he would take the two relatives. Suppose Child 1 and Child 2 were in line and the Mother process arrives. The Father would wait, unless these were the only two children left besides Child 8 and there were no other adults. If there are no more children (besides Child 8) and no more adults, then the Father would take Child 1 and Child 2 on a hayride and then return to wait for Child 8 (so Child 8 and the Mother can go together). If there were Relative processes that hadn't taken any hayrides, one of these adults would need to go with the two children that are left. Possible hayride scenarios include giving rides to groups as follows:

- 2 adults; 3 children; 1 adult and 2 children; 1 adult and 2 children; Mother and Child 8; 2 adults
- 1 adult and 2 children; 1 adult and 2 children; 1 adult and 2 children; 2 adults; 1 adult and 1 child; Mother and Child 8
- Mother and Child 8 and 1 other child; 3 children; 2 adults; 2 adults; 1 adult and 2 children; 1 adult and 1 child
- 1 adult and 2 children; 1 adult and 2 children; 1 adult and 2 children; 2 adults; Mother and 2 children; 1 adult

Your program must print appropriate messages so that your output *clearly indicates* who is doing what. Possible output might appear as follows: (using indentation to see who prints is a good idea)

```
   Child 1 is raking leaves.
      Mother is making the meal.
   Child 2 is raking leaves.
Relative 1 arrives.
Relative 2 arrives.
... (assume all relatives arrive and all children are raking leaves)
   Child 1 is waiting for hayride.
Relative 1 is waiting for hayride.
Relative 3 is waiting for hayride.
   Child 3 is waiting for hayride.
         Father is ready to give hayrides.
         Father gives hayride to Child 1, Child 3 and Relative 1.
   Child 8 is waiting for hayride.
   Child 2 is waiting for hayride.
   Child 1 is sitting at table.
      Mother is waiting for hayride.
         Father gives hayride to Child 2, Child 8 and Mother.
... (assume all have had hayrides and are seated at table)
         Father carves turkey.
   Child 1 is eating.
   Child 3 is eating.
Relative 1 is eating.
      Mother is eating.
... (assume they are all eating)
   Child 1 is done eating.
Relative 4 is done eating.
... (assume they are all done eating)
         Father excuses everyone from table.
Relative 2 goes home.
   Child 7 goes out to play.
       Mother goes to read her book.
         Father goes to watch football.
   Child 1 goes out to play.
...
```

You can use a monitor or semaphores, whichever you prefer. You should add random delay (don't make it too large) at various places so the interleaving of processes is more realistic, for example when children are raking leaves.

Grading

You will be graded on the following:

- Processes must synchronize correctly.
- Processes must follow all rules.
- You must use a delay for activities that take time, such as raking, cooking, giving a hayride, etc.
- Program must compile. Don't hand in programs that won't compile.
- Programs should not deadlock. Write your program one or two processes at a time. Don't put in all processes and expect it to work right the first time! This will help you find any deadlock (or logic) problems in your program.
- All output should be atomic (not interleaved).
- Output should exactly reflect what processes are doing. For the child and relative processes, you must output their number along with their activity (for example, Child 1, Relative 2, etc.). NOTE: BACI has limited stack space for strings. If you get the error message "String table space exhausted" it means you have too many strings in your program and they are too long. If this happens, remove any unnecessary output, abbreviate "Child 1" as "C1", "Relative 1" as "R1", etc. If you still have issues let me know and I can edit the source code to allow for more strings (but it can easily be done without doing this).
- Code must be commented. Points will be taken off if you don't comment appropriately. You should use descriptive variable names and comments throughout the program (not simply at the beginning).

Name this file using your first initial, last name, and "5", all in lower case, for example, jdoe5.cm. If you are working with someone else, please pick one name or the other (not both) to use.