

Computing Lab – I

Assignment 5 (Process Synchronization)

24 August, 2017

Objective of this Assignment

- To understand process synchronization using POSIX semaphores.
- This is an individual assignment.

Semaphore Example

- Used to protect the critical sections of a program (ex. handling shared variables among multiple processes).
- Two functions: wait (P) and signal (V).
- **function** V(semaphore S, integer I):
[S \leftarrow S + I]
- **function** P(semaphore S, integer I):
repeat: [if S \geq I: S \leftarrow S - I **break**]

Semaphore Example

- emptyCount is initially N, fullCount is initially 0, useQueue is initially 1

- **produce:**

```
P(emptyCount)
P(useQueue)
putItemIntoQueue(item)
V(useQueue)
V(fullCount)
```

- **consume:**

```
P(fullCount)
P(useQueue)
item ← getItemFromQueue()
V(useQueue) V(emptyCount)
```

Input to the System

- We want to build up a system for getting summarized information from Twitter based on the personal interests of the user.
- Assume that you have **multiple processes** that can crawl Twitter to collect different tweets and store them in a database (say, in a file)
 - You don't need to implement the actual crawler. You can assume that you already have multiple files having multiple tweets. The crawler processes read those files and put the tweets to a single database file at random times.

Database (File) Format

#PMOIndia Violence in the name of faith is unacceptable.

Deep Learning can tell a gentleman by his shoes **#DeepLearning #TensorFlow**

#MachineLearning #ArtificialIntelligence

India's stature in the world is rising. The world is with us in fighting the menace of terror. I thank all nations helping us doing so: PM **#PMOIndia**

Like <3 or ReTweet this tweet & we will inform you when your fav Nokia 6 is launching on Amazon.in **#UniteFor #Fun #AmazonExclusive**

Microsoft builds autonomous **#AI**-controlled sailplane—able to identify, predict & act on multiple, complex variables <http://msft.social/z6dcnz>

Warm greetings & good wishes to all my fellow citizens, especially my Parsi brothers & sisters, on the Parsi New Year **#PresidentKovind**

Govt to conduct assessment of learning outcomes on Nov 13 **#QualityEducation #NCERT #Transformingeducation**

- The tweets are associated with some hashtags.

User Input and System Output

- Every user connects to the system, and gives a set of hashtags as an input to the system.
- The summarizer process corresponding to every user finds out the tweets related to every hashtag.
- The summarization process returns a summarization of those tweets corresponding to every hashtag, and writes this information in a **single output file** for all the users.

Sample Output

>Enter User:

user123

>Enter hashtags (one at every line, put STOP to stop):

#PMOIndia

#AI

STOP

>Enter User:

User456

>Enter hashtags (one at every line, put STOP to stop):

#MachineLearning

#AI

#EducationForAll

Summary of hashtags for user123 is written at file summary_all.txt

Summary of hashtags for user456 is written at file summary_all.txt

Sample Output

User123 (**Current timestamp: 24/08/2017 11:50 AM**)

#PMOIndia

<summary of tweets>

#AI

<summary of tweets>

=====

User456 (**Current timestamp: 24/08/2017 11:53 AM**)

#MachineLearning

<summary of tweets>

#PMOIndia

<summary of tweets>

#EducationForAll

<summary of tweets>

=====

**Note that summary for
different users for the same
hashtag may be different**

Process Flow

- The crawler processes crawl (read) the tweet files at random time and write the data in a database file tweetdata.txt. **The number of crawler processes will be an input from the user.**
- The parent summarization process keeps on listening for the inputs (user name and hashtags) from the users.
- Once it gets the input from the user, it creates a child summarization process to handle the summarization.
- The child process filters our relevant tweets, summarize them, and generates the output file.

Submission Instructions

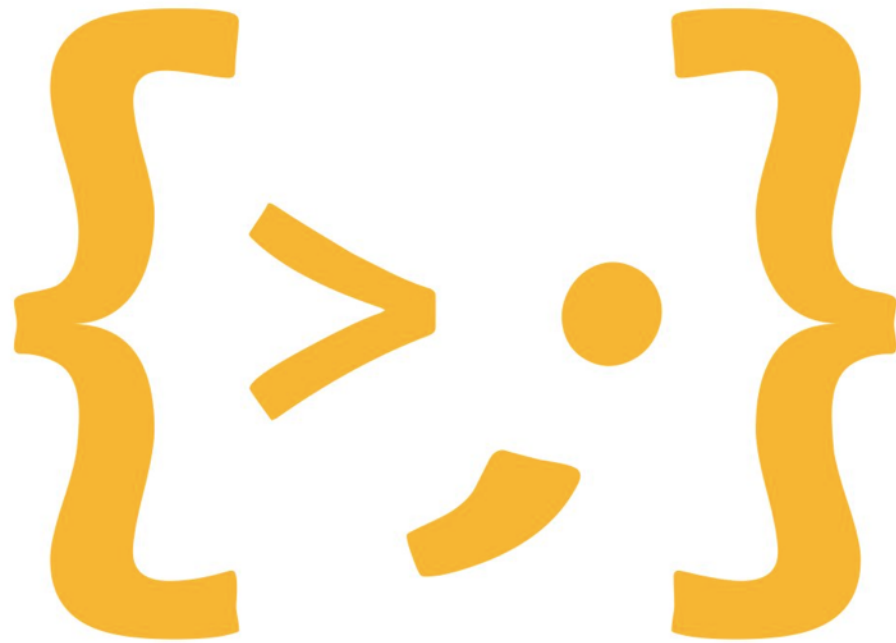
- The code needs to be written in C or C++. You can use the solution for Assignment 3 for summarization. Submit the C/C++ source in a single compressed file. The file must contain the name and roll number of the student.
- **Submission Deadline – September 07, 2:00 PM IST**

Marking scheme

- Crawler process synchronization: 30%
- Reader writer synchronization between crawler processes and the summarization processes: 30%
- Handling synchronization at child summarizer during output production: 30%
- Documentation, coding, understandability: 10%

Some Useful Links

- https://linux.die.net/man/7/sem_overview
- <http://www.csc.villanova.edu/~mdamian/threads/posixsem.html>
- http://www.minek.com/files/unix_examples/semab.html



HAPPY

CODING