

# Design patterns

ממשו את תבניות העיצוב הבאות :

1. ממש **תור מוגן טרדים**. טרד שממתיך לשחרור איבר מהתור כאשר אין איברים ימתין על cond.

עליכם לממש 4 פונקציות :

- א. createQ - מייצר תור חדש
- ב. destoryQ - הורס תור
- ג. enQ (מכניס איבר חדש לתור קיים Q)
- ד. deQ (מוציא איבר מתור Q)

בסעיף זה - אפשר להשתמש ב++C אם רוצים.

אפשר להניח - כל הקלטים חוקיים. אם מבקשים להוציא איבר מהתור יש בו איברים. פרט לזה לא צריך להניח שום הנחה נוספת. איבר (והתור עצמו) ייוצגו בעזרת void\* (תוכלו לעשות casting למה שתרצו)

2. השתמשו בתור משאלה (1) - לממש **active\_object**. ה active object צריך לקבל פוינטר לפונקציה שתקרא עבור כל איבר בתור ופונקציה שתקרא כאשר הטיפול מסתיים.

עליכם לממש 2 פונקציות:

- א. newAO - יקבל 3 פרמטרים - תור, 2 מצביעים לפונקציות-פונקציה שתקרא לטיפול בכל איבר ופונקציה שתקרא לאחר הטיפול (לדוגמא לצורך מימוש pipeline)
- ב. destroyAO - יקבל AO ותהרוס אותו (תקרא ל pthread\_cancel ותשחרר את כל מבני הנתונים)

בסעיף זה - אפשר להשתמש ב++C אם רוצים.

3. **בנה pipeline של active\_object מסעיף (2)**

- א. ה active\_object הראשון יקרא קלט שמגיע ל socket בעזרת השרת מתרגיל 3. (ממטלה 3)
- ב. כל מחזורת המגיעה מהמשתמש תכנס לשרשרת של active\_object. המחזורת תכיל אותיות באנגלית בלבד (גדולות או קטנות) ללא רווחים.
- ג. ה Active\_object הראשון יפעיל צופן קיסר בהיסט של 1 על כל קלט שהמשתמש מכניס. שים לב שלא נהפוך אות גדולה לקטנה. (לדוגמא HAL יהפוך לIBM) טפלו במקרי הקצה.
- ד. ה active\_object השני יהפוך אותיות קטנות לגדולות וההפך.
- ה. ה Active\_object האחרון יחזיר תשובה למשתמש - המשתמש יקבל תשובה למחזורת שהוא שלח ורק אותה. אסור להחזיר למשתמש תשובה למרוזת שהוא לא שלח.

אתם רשאים לאחסן מידע נוסף בתור (אם אתם בוחרים בכך)

בסעיף זה - אפשר להשתמש ב++C אם רוצים

4. ממש **Guard** (או ScopeMutex) - צריך להשתמש ב++C

בנה אפליקציית בדיקה המאפשרת רק לטרד אחד להכנס לפונקציה המעדכנת מצביע גלובלי. האם ניתן להשתמש בguard כזה כדי להגן על strtok? (במקום strtok\_r) כתוב את התשובה בהערות.

5. ממש **singleton** בעזרת נעילות של posix. צריך להשתמש ב++C ובtemplate

עליכם להגדיר 2 פונקציות :

# Design patterns

א. Instance - שמחזירה מצביע לסינגלטון.  
ב. Destroy - שהורסת אותו.  
בנה אפליקציית בדיקה שממפה קובץ לזיכרון בתור singleton.

## 6. בנה Reactor.

הReactor ימומש בעזרת poll או select לבחירתכם ובעזרת טרד אחד או יותר לבחירת המשתמש. ממש את הפונקציות הבאות:

- א. newReactor - (מחזיר void \*) יוצר Reactor חדש  
ב. InstallHandler - יקבל מצביע לReactor, מצביע לפונקציה וfile descriptor - יתקין פונקציה שתטפל בקלט בלבד על הfile descriptor  
ג. RemoveHandler - יקבל מצביע לReactor ישחרר handler לFD  
ממש את selectserver (או pollserver) של beej (פרק 7.2 או 7.3) בעזרת reactor שבנית. בנה גם selectclient (קליינט התומך בצאט) שידבר עם הסרבר שבנית לצרכי בדיקה.

הגישו את כל הקוד (של כל הסעיפים) **כספריה משותפת** (shared object) כאשר תוכניות הבדיקה משתמשות בספריה.

לקומפילציה של ספריה משותפת השתמשו ב  
gcc --shared -fPIC -pthread

הגישו makefile הבונה את הספריה וכל תוכניות הבדיקה.

גם תרגיל זה יבדק בעזרת moss

תוכלו לצרף קובץ readme.txt עם כל ההנחות. – מומלץ מאוד !!