

מטלה 6 – תבניות עיצוב

שם: ניב

משפחה: קוטיק

ת.ז.: 208236315

שם המרצה: חגיז מרדכי

תאריך הגשה: 03.06.22

קבצים:

client.c – לקוח לצורך בדיקות במידת הצורך

Active_object – main1.c

main1.h

main1_test.c - לצורך בדיקות של Active_object

Guard – guard.cpp (יש לו גם main לצורך test)

guard.hpp

Singleton – singleton.cpp (יש לו גם main לצורך test)

reacto – reactor.cpp

reactor.hpp

selectclient_reactor_test.cpp - לצורך בדיקות של reactor

Makefile (הכולל shared object)

readme.pdf (יש גם סרטון הרצה)

איר להריץ:

בשביל לקמפל את כל הקבצים יש לכתוב `make all`.

בשביל להפעיל את `main1` יש לכתוב `make Active_object` - יש להשים לב כי התוכנית רצה לאחר `make Active_object`.

בשביל להפעיל את הלקוח לצורך בדיקה עם השרת של `main1` יש לכתוב `make Client` - יש להשים לב כי התוכנית רצה לאחר `make Client`.

בשביל להפעיל את הבדיקה של `Active object` יש לכתוב `make Main1_test` - יש להשים לב כי התוכנית רצה לאחר `make Main1_test`.

בשביל להפעיל את `guard` יש לכתוב `make guard` - יש להשים לב כי התוכנית רצה לאחר `make guard`.

בשביל להפעיל את ה `singleton` יש לכתוב `make Singleton` - יש להשים לב כי התוכנית רצה לאחר `make Singleton`.

בשביל להפעיל את `reactor` יש לכתוב `make Reactor` – יש להשים לב כי התוכנית רצה לאחר `make Reactor`.

בשביל להפעיל את הבדיקה של `reactor` יש לכתוב `make Reactor_test` - יש להשים לב כי התוכנית רצה לאחר.

בשביל למחוק את כל קבצי ההרצה - `make clean`.

*** יש להשים לב כי הקומפליציה נעשית על ידי ספריות משותפות `shared object` ***

קובץ `main1` של `active object`:

על ידי `make Active_object` מפעילים את ה `active object` (קובץ `main1`). לאחר מכן ניתן לחבר את הלקוח על ידי פעולת `make client`.

בשביל לנתק את השרת באופן בטוח יש להקליד בשורת הפקודה של השרת את המילה - `YES`.

קובץ לקוח:

על ידי `make client` מפעילים את הלקוח לאחר שהשרת מחובר. בשביל לנתק את הלקוח השרת באופן בטוח יש להקליד בשורת הפקודה `exit`.

קובץ טסטים של `active object`:

על ידי `make Main1_test` מפעילים את קובץ הטסים. בקובץ הטסים יש הפעלה של השרת ושני לקוחות באותו הזמן. הלקוחות מבצעים פעולות בו זמנית וניתן לראות את הקובץ הקלט והפלט של כל לקוח לאחר `active object`.

קובץ guard:

על ידי make Guard מפעילים את הקובץ. בנוסף יש לו ב guard.cpp פונקציית main לצורך בדיקה. בבדיקה ניתן לראות הרצה של 2 תהליכים כאשר שניהם ניגשים למשתנה גלובלי ובעזרת pthread_mutex_t מאפשרים רק לתהליכון אחד להיכנס כל פעם ולבצע את הפעולה על המשתנה הגלובלי.

קובץ Singleton:

על ידי make Singleton מפעילים את הקובץ. בנוסף יש לו ב Singleton.cpp פונקציית main לצורך בדיקה. בבדיקה ניתן לראות כי אנו יוצרים שני אובייקטים שונים משני בלוקים שונים וניתן לראות כי האובייקט לא השתנה לאחר יצירה של השני. כלומר אין הקצאה חדשה של זיכרון. בסוף ניתן לראות את פעולת המחיקה רק פעם אחת.

קובץ Reactor:

על ידי make Reactor מפעילים את הקובץ. לאחר מכן ניתן לחבר את הלקוח על ידי פעולת make client.

בשביל לנתק את השרת באופן בטוח יש להקליד בשורת הפקודה של השרת את המילה - YES.

בקובץ הנ"ל ניתן לראות את select על ידי מימוש Reactor.

דוגמא להרצת התוכנית main1:

השרת: בתמונה הנ"ל ניתן לראות את הרצת השרת. השרת מחכה ללקוח וכאשר לקוח מתחבר אזי השרת מדפיס את ה ip. בנוסף לכך אם רוצים לנתק את השרת ניתן להקליד בחלון השרת YES. (בתמונה הנ"ל ניתן לראות כי התחברו לשרת שני לקוחות)

```
kotek@kotek:~/frome-win/C_EX6$ ./main1
server: waiting for connections...
Do you want to get out of the server (YES OR NO)?
server: got connection from 127.0.0.1
server: got connection from 127.0.0.1
□
```

לקוח: ניתן לראות לאחר התחברות הלקוח מודפס הכתוב Enter the string. כלומר הלקוח מוכן לקבל פקודות אשר ישלח לשרת דרך פקטה.

```
nivk@kotekN:/mnt/c/Users/kotek/Desktop/c/c_EX5$ make client
gcc -o client client.cpp
./client
Socket successfully created..
connected to the server..
Enter the string : □
```

בדיקות:

בתמונה הנ"ל ניתן לראות את הרצת התוכנית של main1. ניתן לראות את התחברות השרת ושני הלקוחות. כמו כן ניתן לראות את כי שני הלקוחות שולחים כל מיני מילים והשרת מבצע עליהם פעולות ולבסוף שולח להם תשובה לאחר הפעולות שקיבל בסוקט. לבסוף יש התנתקות הן של הלקוחות והן של השרת.

```
kotek@kotek:~/frome-win/C_EX6$ make Main1_test
./main1_test
server: waiting for connections...
Do you want to get out of the server (YES OR NO)?
server: got connection from 127.0.0.1
OUTPUT: bBcCdD
From client: AaBbCc      To client : server: got connection from 127.0.0.1
OUTPUT: bBcCdD
OUTPUT: oJw
OUTPUT: oJw
OUTPUT: lPUf1
OUTPUT: lPUf1
OUTPUT: qff1
OUTPUT: qff1
OUTPUT: IJHLMNOP
OUTPUT: IJHLMNOP
OUTPUT: bcdefgh
OUTPUT: bcdefgh
From client: AaBbCc      To client : From client: NiV      To client : From client: NiV      To client : From client: KoteK To cli
ent : From client: KoteK      To client : From client: PEEK      To client : From client: PEEK      To client : From client: highl
mnop      To client : From client: highlmnop      To client : From client: ABCDEFG      To client : From client: ABCDEFG
To client : Server Exit...
Client Exit...
Server Exit...
Client Exit...
server: got connection from 127.0.0.1
main thred exits
Client Exit...
```

דוגמא להרצת התוכנית guard:

בתמונה הנ"ל ניתן לראות את הרצת התוכנית של guard. ניתן לראות את כי מופעל 2 תהליכונים על משתנה גלובלי 1 וכל פעם רק תהליכון אחד נכנס ומבצע את הפעולה על המשתנה הגלובלי. לבסוף יש הדפסה של מספר התהליכון.

```
kotek@kotek:~/frome-win/C_EX6$ make Guard
./guard
Thread 1 : 1
Thread 1 : 2
Thread 1 : 3
Thread 1 : 4
Thread 1 : 5
Thread 1 : 6
Thread 1 : 7
Thread 1 : 8
Thread 1 : 9
Thread 1 : 10
0
Thread 2 : 11
Thread 2 : 12
Thread 2 : 13
Thread 2 : 14
Thread 2 : 15
Thread 2 : 16
Thread 2 : 17
Thread 2 : 18
Thread 2 : 19
Thread 2 : 20
Thread 1 returns: 139939124217600
Thread 2 returns: 139939115763456
kotek@kotek:~/frome-win/C_EX6$
```

דוגמא להרצת התוכנית Singleton:

בתמונה הנ"ל ניתן לראות את הרצת התוכנית של Singleton. ניתן לראות את כי יש כניסה רק פעם אחת ל constructor ופעם אחת ל destructor למרות שיש יצירה פעמים של אובייקטים. בנוסף ניתן לראות כי יש פעמים הדפסה של in use המראה כי זה אותו אובייקט Singleton. לבסוף ניתן לראות את destructor.

```
kotek@kotek:~/frome-win/C_EX6$ make Singleton
./singleton
Entering main()
Test::constructed
Test::in use
Test::in use
Leaving main()
Singleton::MemGuard::~~MemGuard::destroyed
Test::destroyed
```

דוגמא להרצת התוכנית Reactor:

בתמונה הנ"ל ניתן לראות את הרצת התוכנית של Reactor.

השרת: בתמונה הנ"ל ניתן לראות את הרצת השרת. השרת מחכה ללקוח וכאשר לקוח מתחבר אזי השרת מדפיס את ה ip. בנוסף לכך אם רוצים לנתק את השרת ניתן להקליד בחלון השרת YES. (בתמונה הנ"ל ניתן לראות כי התחברו לשרת שני לקוחות)

```
kotek@kotek:~/frome-win/C_EX6$ make Reactor
./reactor
DO you wat tp get out of the server? (YES OR NO)?
selectserver: new connection from 127.0.0.1 on socket 4
selectserver: new connection from 127.0.0.1 on socket 5
□
```

בדיקות:

בתמונה הנ"ל ניתן לראות את הרצת התוכנית של הבדיקות של Reactor. ניתן לראות את התחברות השרת ושני הלקוחות. כמו כן ניתן לראות את כי שני הלקוחות שולחים הודעות כמו בצאט. כלומר אחר שולח הודעה ובצד השני מתקבל את הלקוח האחר ההודעה שאותו לקוח שלח. לבסוף יש התנתקות הן של הלקוחות והן של השרת.

```
kotek@kotek:~/frome-win/C_EX6$ make Reactor_test
./selectclient_reactor_test
DO you wat tp get out of the server? (YES OR NO)?
selectserver: new connection from 127.0.0.1 on socket 4
selectserver: new connection from 127.0.0.1 on socket 5
OUTPUT --: from omer::Hello, my name is omer
OUTPUT --: from niv::Hello, my name is Niv
OUTPUT --: from omer::What's going on?
OUTPUT --: from niv::What's going on?
OUTPUT --: from omer::Ok thank you
OUTPUT --: from niv::Ok thank you
OUTPUT --: from omer::what day is it today
OUTPUT --: from niv::what day is it today
OUTPUT --: from omer::higklmnop
OUTPUT --: from niv::higklmnop
OUTPUT --: from omer::what is the time now
OUTPUT --: from niv::what is the time now
OUTPUT --: exit
Client Exit...
selectserver: socket 5 hung up
Client Exit...
selectserver: socket 4 hung up
selectserver: new connection from 127.0.0.1 on socket 4
Client Exit...
```

הערות:

- א. השרת רץ על ידי פורט 8584
- ב. בסיום התוכנית מומלץ לכתוב `make clean`
- ג. מצורף גם לכל אחד בדיקות.
- ד. במידה והפורט בשימוש של מערכת ההפעלה מומלץ להחליף את מספר הפורט.
- ה. הן השרת והן הלקוח רצים על `localhost`
- ו. מומלץ לבדוק לפני ההרצה שכל הקבצים נמצאים.
- ז. חלק מהקבצים בסיומת `c` וחלק ב `cpp`

:Makefile:

בקובץ ה `makefile` ניתן לראות בנייה של ספריות משותפות. מכיוון של תבנית עיצוב עומדת בפני עצמה קיים לכל אחד ספרייה משלה. ניתן לראות בקובץ וגם בקימפול את בניית כל הספריות. (קיים קומפילציה עם הרבה קבצים).