

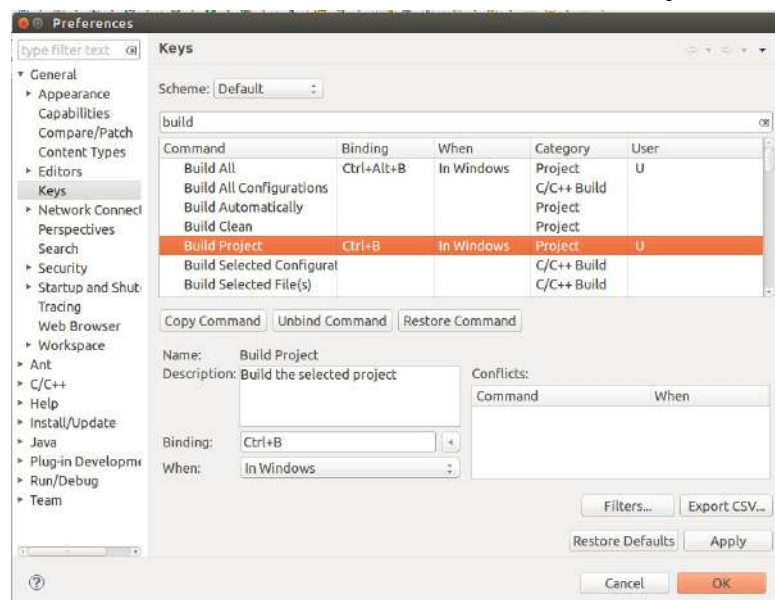
# הנחיות לשימוש ב XV6

## שלבם מקדימים:

1. הורידו את תמונת Ubuntu 16.04 המופיעה בהודעות אתר הקורס. הפעילו אותה באמצעות Vmware או Virtualbox כפי שמוסבר באתר. שימו לב של Vmware אין גרסא חנימית ל Mac.
2. פתחו shell.
3. התקינו qemu:  
`sudo apt-get install qemu`
4. התקינו expect:  
`sudo apt-get install expect`
5. הורידו ופתחו את התיקיה המכילה את גרסת xv6 הרלוונטית לממן.
6. פתחו את קובץ Makefile ושנו את השורה:  
`debug ?= false` ל `debug ?= true`.
7. צרו קובץ בשם gdbinit, עם התוכן הבא כרשום בהנחיות:  
`set auto-load safe-path /`

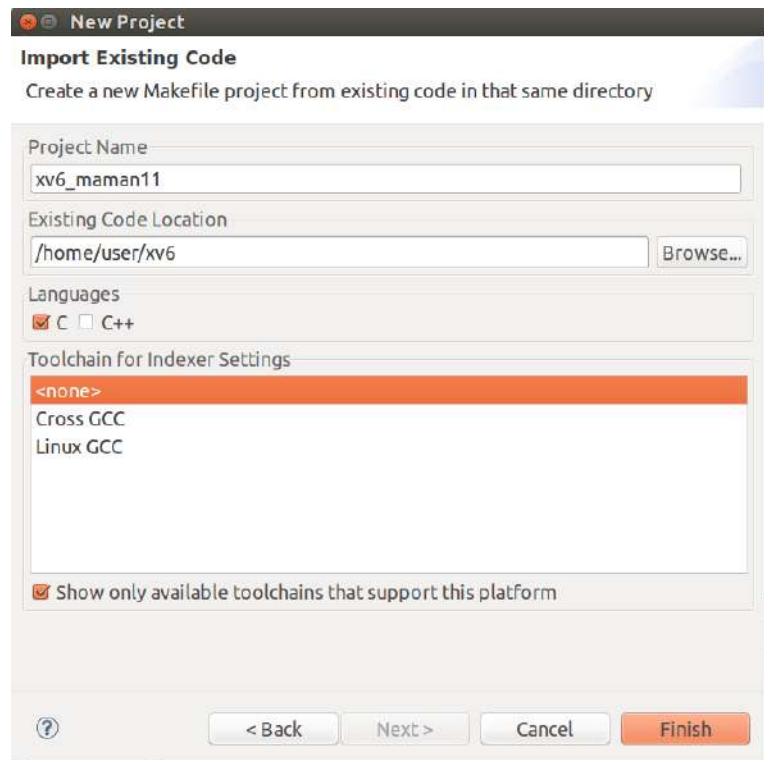
## הכנת סביבת אקליפס:

- הגדרת קיצור מקשים לבניית פרויקט נוכחי במקום כולם:
- 1. בשורת הכותרת, לחצו על Preferences → Window.
- 2. פתחו את קטגוריית General → Keys.
- 3. חפשו את פקודות Build בפס החיפוש.
- 4. ב Build All, תחת Binding, מחקו את הצירוף Ctrl+B, והוסיפו צירוף כגון Ctrl+Alt+B.
- 5. ב Build Project הוסיפו את הצירוף Ctrl+B.



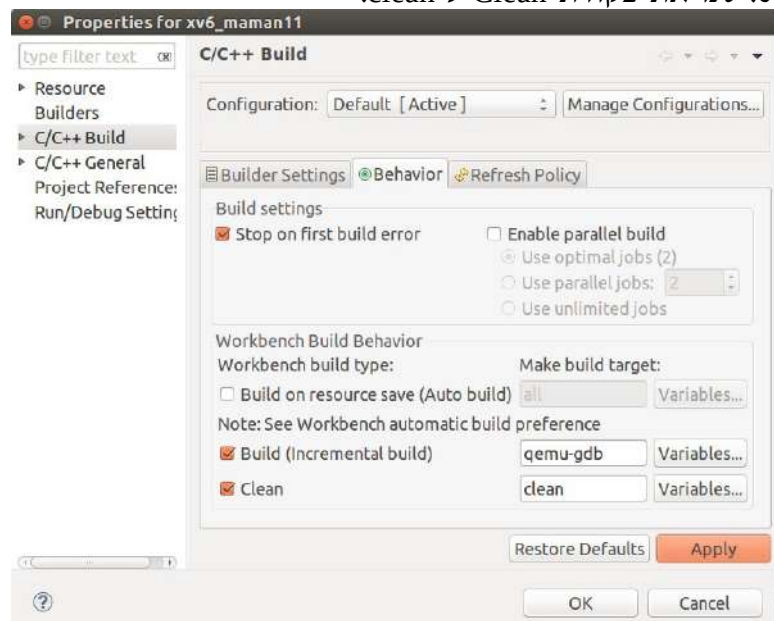
– ייבוא תיקיית xv6 כפרויקט אקליפס:

1. בשורת הכותרת, לחצו על Import → File.
2. בחרו את האופציה Existing Code as Makefile Project → C/C++.
3. תנו שם לפי הממך, ונווטו אל תיקיית xv6 שנפתחה. כבו סימון שפת ++C ובחרו את אופציית ללא Toolchain. לחצו על Finish.



– איחוד פקודות בנייה (ניקוי אוטומטי והפעלת qemu + דיבאג):

1. לחיצה ימנית על הפרויקט שנוצר, בחרו את Properties בתפריט.
2. בחרו את האופציה C/C++ Build.
3. תחת טאב Builder Settings שנפתח, כבו סימון Use default build command ושנו את פקודת Build ל make clean.
4. בחרו את טאב Behavior.
5. שנו את פקודת Build (Incremental build) ל qemu-gdb.
6. שנו את פקודת Clean ל clean.

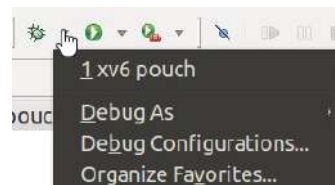


## הרצת דיבאג באקליפס (על pouch.c לדוגמא), אחרי השלבים המקדימים:

0. עברו לפרספקטיבת C או Debug, פתחו את קובץ pouch.c של הפרויקט, בחרו נקודות עצירה ושנו את הקוד במידת הצורך.

1. אם שנייתם משהו בקוד, בנו פרויקט (Ctrl+B) וסגרו את החלון השחור שנפתח (או בטלו בנייה) – פעולה זו מקמפלת מחדש את הקוד בין השאר.

2. פתחו את תפריט צלמית הדיבאג, בחרו Debug Configurations.



3. הוסיפו קונפיגורציית דיבאג תחת C/C++ Remote Application.

4. בחרו את הקונפיגורציה שנוצרה, ועדכנו בה את הפרמטרים הבאים:

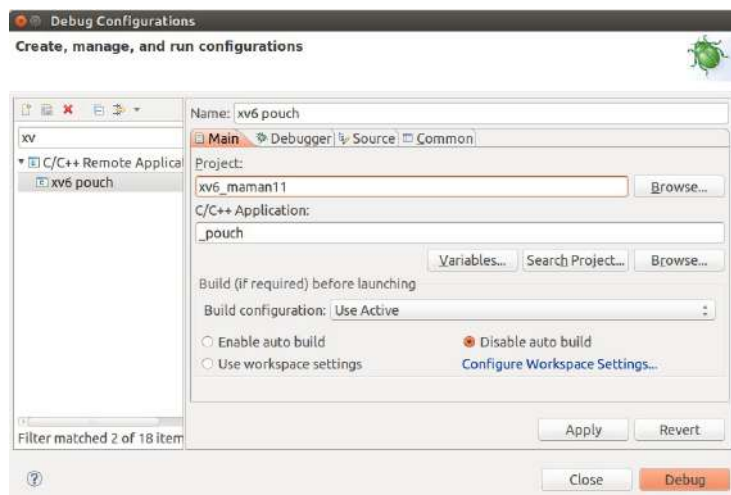
תחת טאב Main

– שם משמעותי כגון ממך + pouch.

– אפליקציה = pouch\_.

– Build Configuration = Use Active

– Disable Auto Build

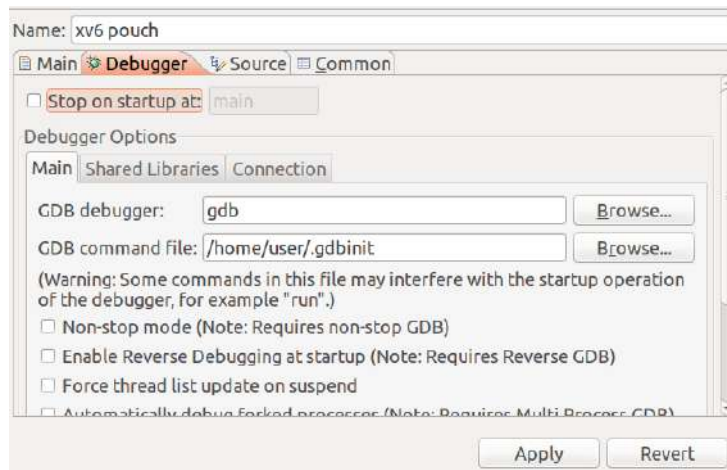


תחת טאב Debugger, תת טאב Main

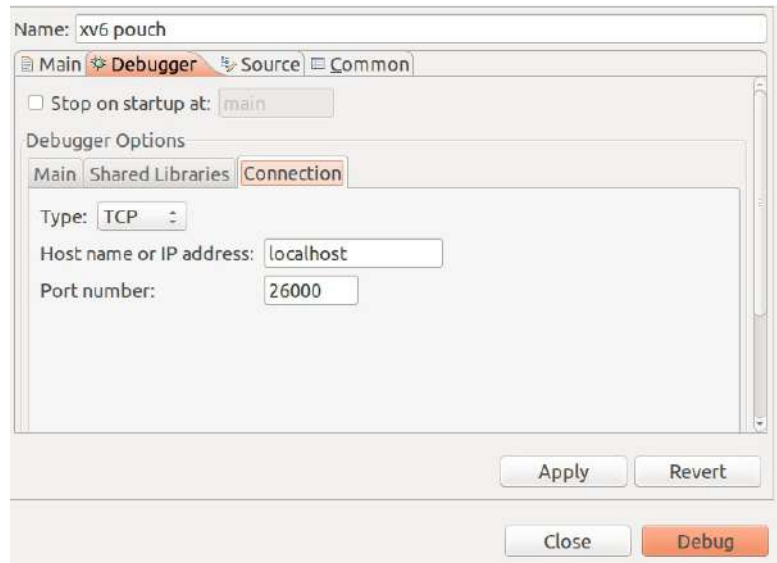
– כבו סימון Stop on startup at main

– GDB command file = קובץ gdbinit (הוא מוסתר, אז צריך להציג אותו באמצעות לחיצה

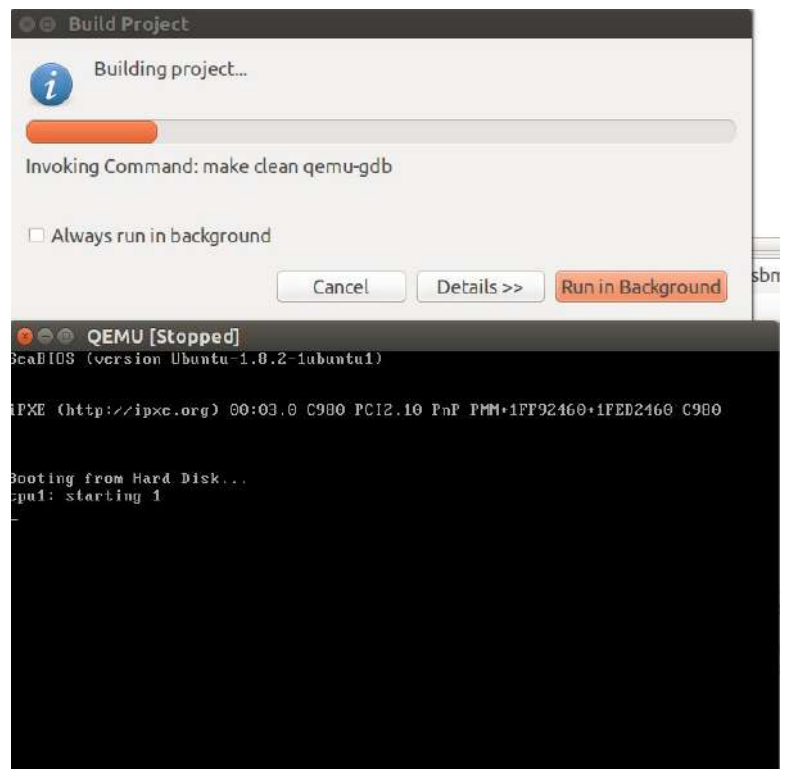
מינית ← Show Hidden Files).



תחת טאב Debugger, תת טאב Connection –  
Port number = 26000



5. הריצו דיבאג (בהמשך הוא יופיע תחת רשימת הקונפיגורציות המועדפות) – הוא יתחיל וייכנס למצב האזנה (המתנה).
6. בנו פרויקט (Ctrl+B) במקביל לדיבאג – אם הכל עבד, אחרי מספר שניות הטרימינל של qemu צריך לעבור למצב קבלת פקודות, או שהדיבאג של אקליפס יעצר בנקודה שהוגדרה (בקרנל).  
עצירת הדיבאג גם סוגרת את qemu.  
שימו לב שכל שינוי בקוד או בנקודות עצירה דורש הפעלה מחדש של שלבים 1+5+6.



## הרצת בדיקות על פתרון ביי"ס (עבור pouch לדוגמא):

1. פתחו shell.
  2. נווטו לתוך תיקיית xv6:  
`cd {xv6_path}`
  3. הריצו פקודה:  
`make clean`
  4. הריצו פקודה:  
`make`
  5. העתיקו את קבצי הפתרון לתיקיית xv6.
  6. הריצו פקודת `touch` על כל קובץ:  
`touch _pouch pouch.o`
  7. הריצו פקודה:  
`make qemu`
- כל הבדיקות והניסויים מתבצעים דרך הטרמינל. אין שימוש באקליפס.

## הרצת בדיקות expect:

כרשום בהנחיות (צפוי זמן ריצה ארוך).

## טיפים:

- חלק מהפקודות שונות מ `stdio`. החתימה של חלק מהן נמצאת ב `user.h`.
- בפרט, יישום חלקי של `printf` הסטנדרטית מוגדר ב `console.c`, תחת השם `cprintf` - אפשר להפעיל אותה מתוך הקרנל.