# CVE-2016-5195



## מהלך הסקירה

### מושגים בסיסיים

אני אתחיל את הסקירה בהסבר לגבי כמה מושגים בסיסיים, ואז אעבור על קטע קוד שמנצל את החולשה ואסביר עליו. אם בכל שלב יש לכם שאלות אתם מוזמנים להרים את היד ואשתדל לענות כמה שיותר.

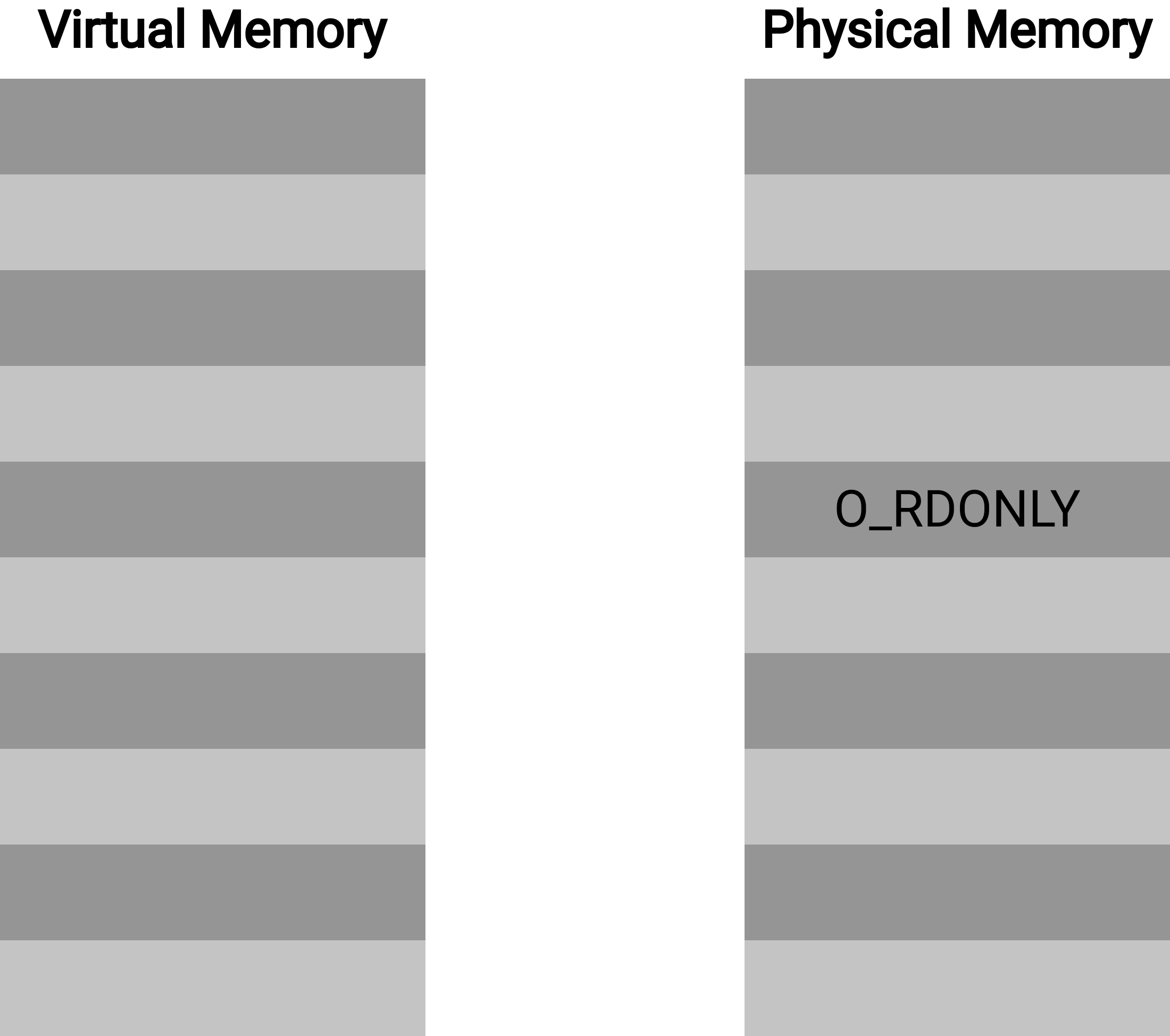
סגור, אז לפני שנתחיל, כדי להבין את החולשה צריך להכיר שני מובנים בסיסיים.

Copy on write – שיטת ניהול זיכרון שמאפשרת ליצור עותק פרטי של דפים בזיכרון בעת כתיבה אל טווח כתובות ממופה.

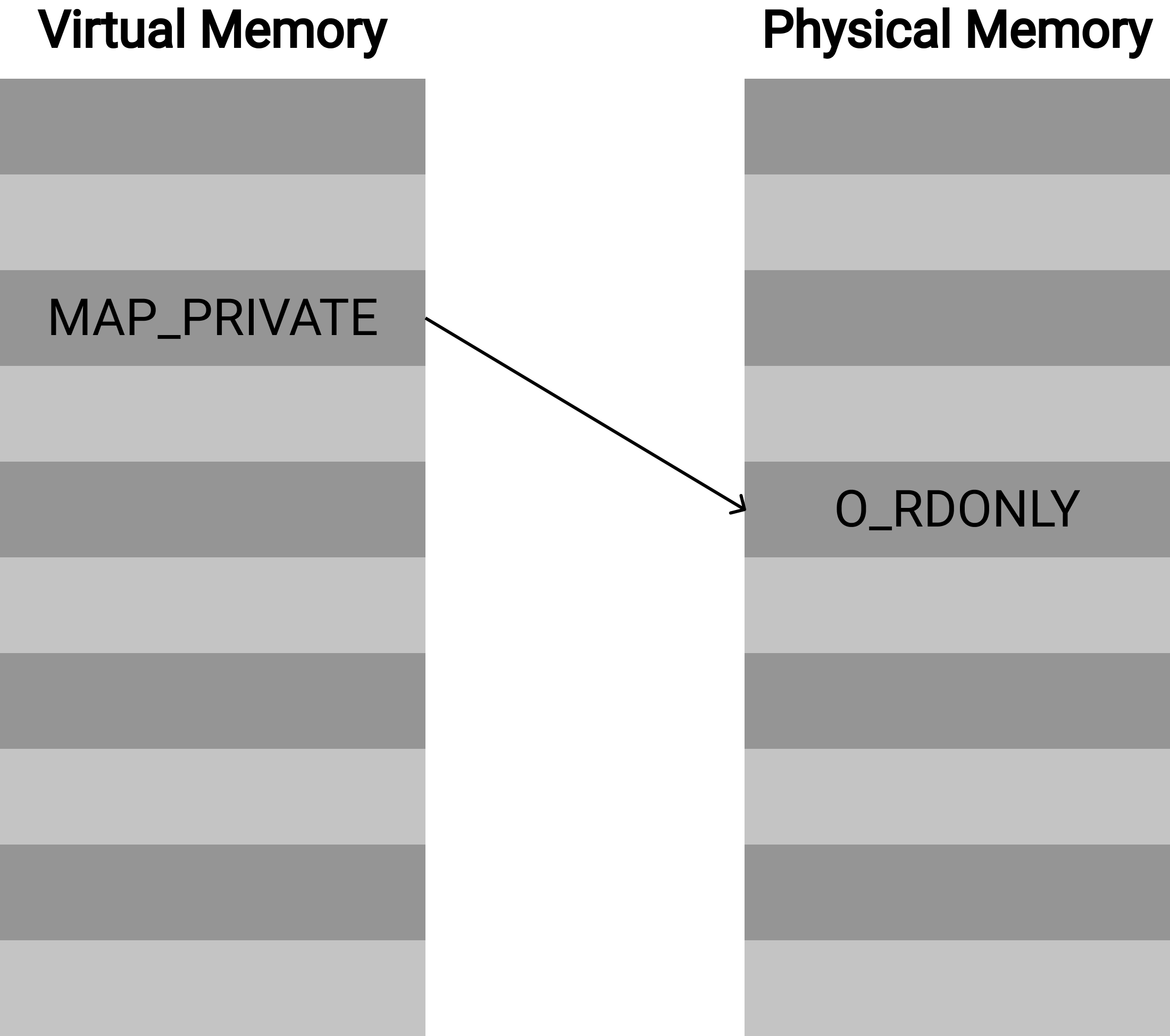
Race condition – סוג חולשה שמנצל context switching בין שני thread-ים.

ארחיב קצת על COW לפני שאמשיך. נגיד וטעון לי קובץ חשוב לזיכרון בהרשאות קריאה בלבד. תהליך שרץ יכול למפות את הקובץ למרחב הכתובות הווירטואלי שלו. כמובן שהתהליך לא יכול לשנות את תוכן הקובץ המקורי, אבל מה אם הוא רוצה לבצע שינויים זמניים שתקפים רק לעצמו? בעיקרון לא אמורה להיות שום בעיה, ומערכת ההפעלה באמת מאפשרת את זה. ניתן למפות את הקובץ למרחב הכתובות הווירטואלי עם הדגל MAP\_PRIVATE, מה שמאפשר את השימוש בCOW. מנגנון ה COW אומר שכל עוד התהליך רק קורא מאותן הכתובות, המיפוי יצביע לעותק המקורי שלהן, אבל ברגע שתהליך יבצע כתיבה, הוא ייצור את העותק וישנה את המיפוי של הכתובות הווירטואליות אליו. רק לאחר מכן הוא יכתוב לתוכו.

Open O\_RDONLY

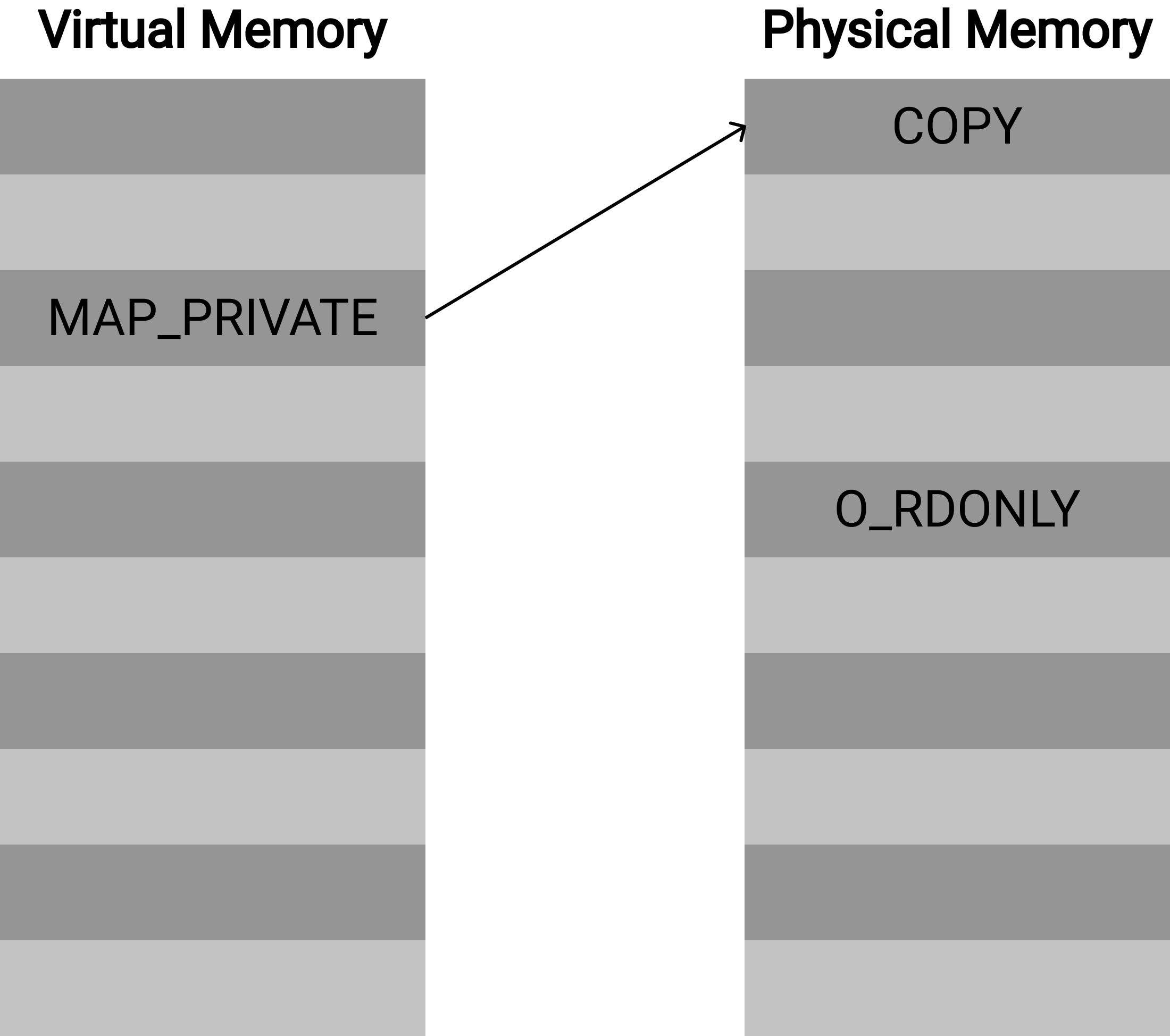


Mmap MAP\_PRIVATE

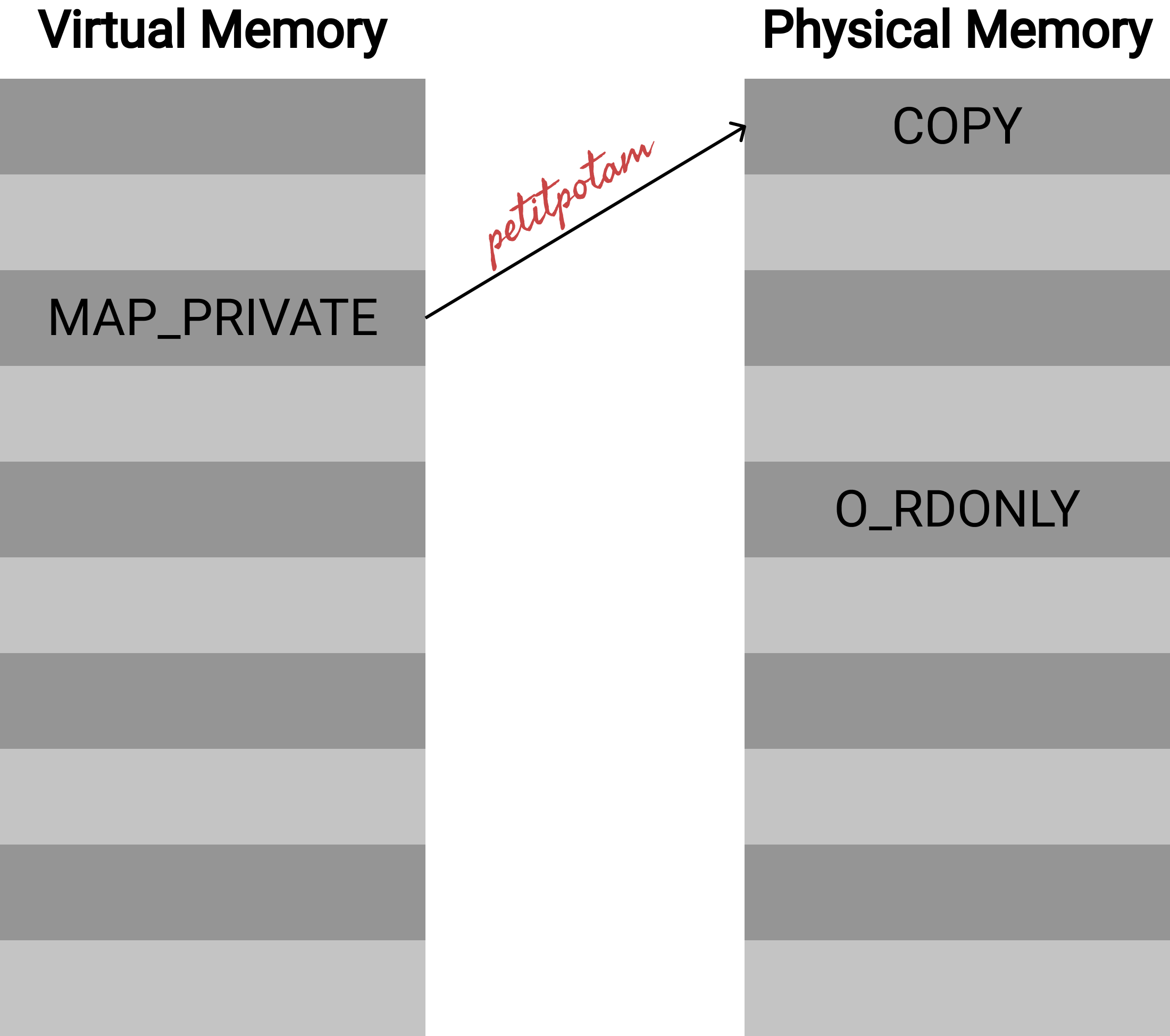


WRITE

קודם כל נוצר המיפוי לעותק החדש



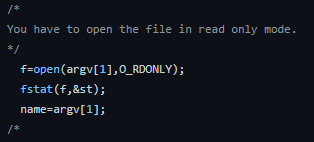
ובשלב נפרד מתבצעת הכתיבה.



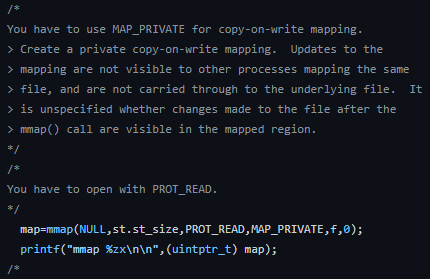
יש שאלות?

### הקוד

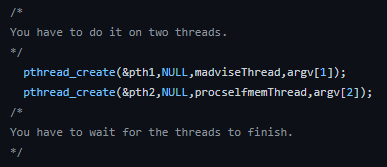
בשלב הראשון, הקוד טוען את הקובץ הרצוי בהרשאות קריאה בלבד אל הזיכרון.



אחר כך, מתבצע מיפוי של הקובץ אל התהליך, אפשר לראות שמועבר גם הדגל MAP\_PRIVATE על מנת לאפשר COW.

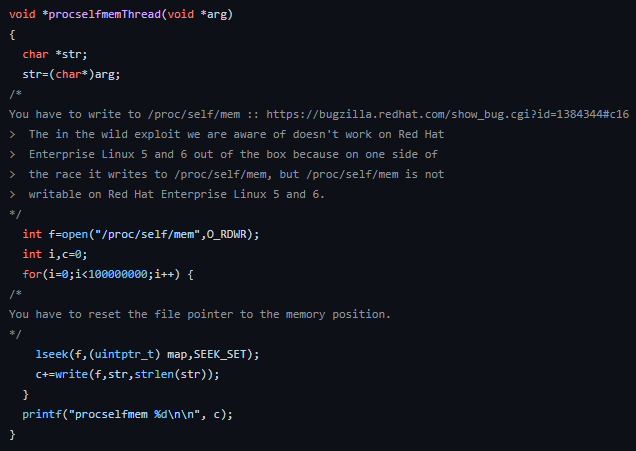


ואז מורצים שני thread-ים במקביל.



זו הפונקציה הראשונה. היא דואגת לפתוח את הקובץ /proc/self/mem לכתיבה וקריאה. זה קובץ מיוחד במערכת הקבצים שמייצג את הזיכרון הווירטואלי של התהליך הנוכחי שרץ. שינוי שנעשה בקובץ הזה יתרחש בעצם על הזיכרון הווירטואלי של התהליך.

לאחר מכן הפונקציה נכנסת ללולאה בה הוא מנסה כל הזמן לכתוב למיפוי של הקובץ שנוצר קודם. אפשר לראות כאן שבתוך הלולאה, נעשה seek לכתובת של map שהוא המיפוי הפרטי של הקובץ, ואז נכתב לתוכו התוכן. הכתיבה לכתובת בקובץ mem שווה ערך לכתיבה ישירות למיפוי.

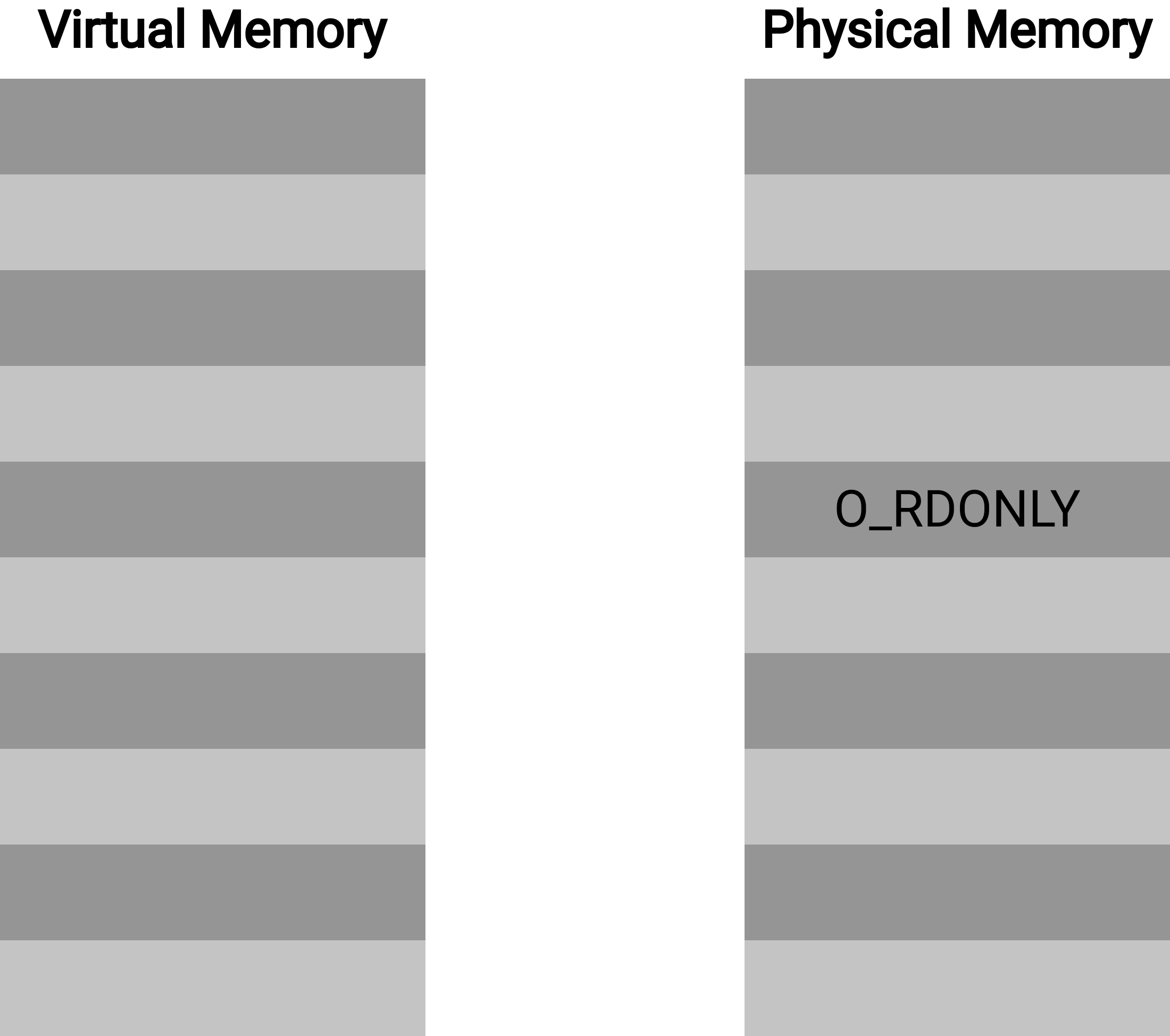


זו הפונקציה השנייה, היא קוראת בלולאה לsyscall – madvice עם הגדל MADV\_DONTNEED. הsyscall הזה מודיע שאין יותר צורך בעותק הפרטי ומשנה את המיפוי חזרה לקובץ המקורי בזיכרון.

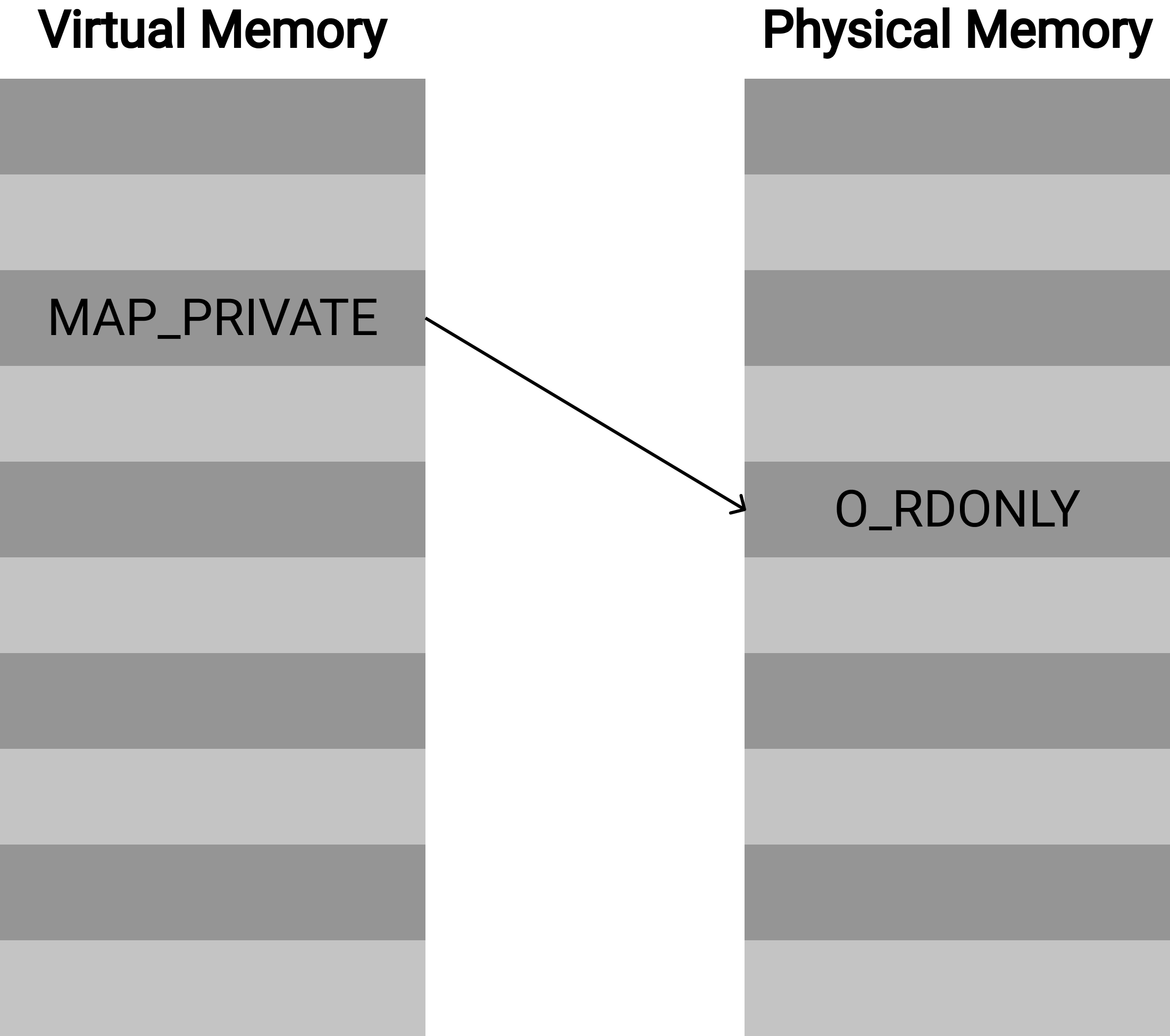


אעבור מהר עכשיו שוב על שתי הפונקציות, ואני רוצה שאחר כך מישהו ינסה להגיד לי מהו הrace condition שמאפשר כתיבה לקובץ המקורי.

קודם כל נטען קובץ בהרשאות כתיבה בלבד לזיכרון.



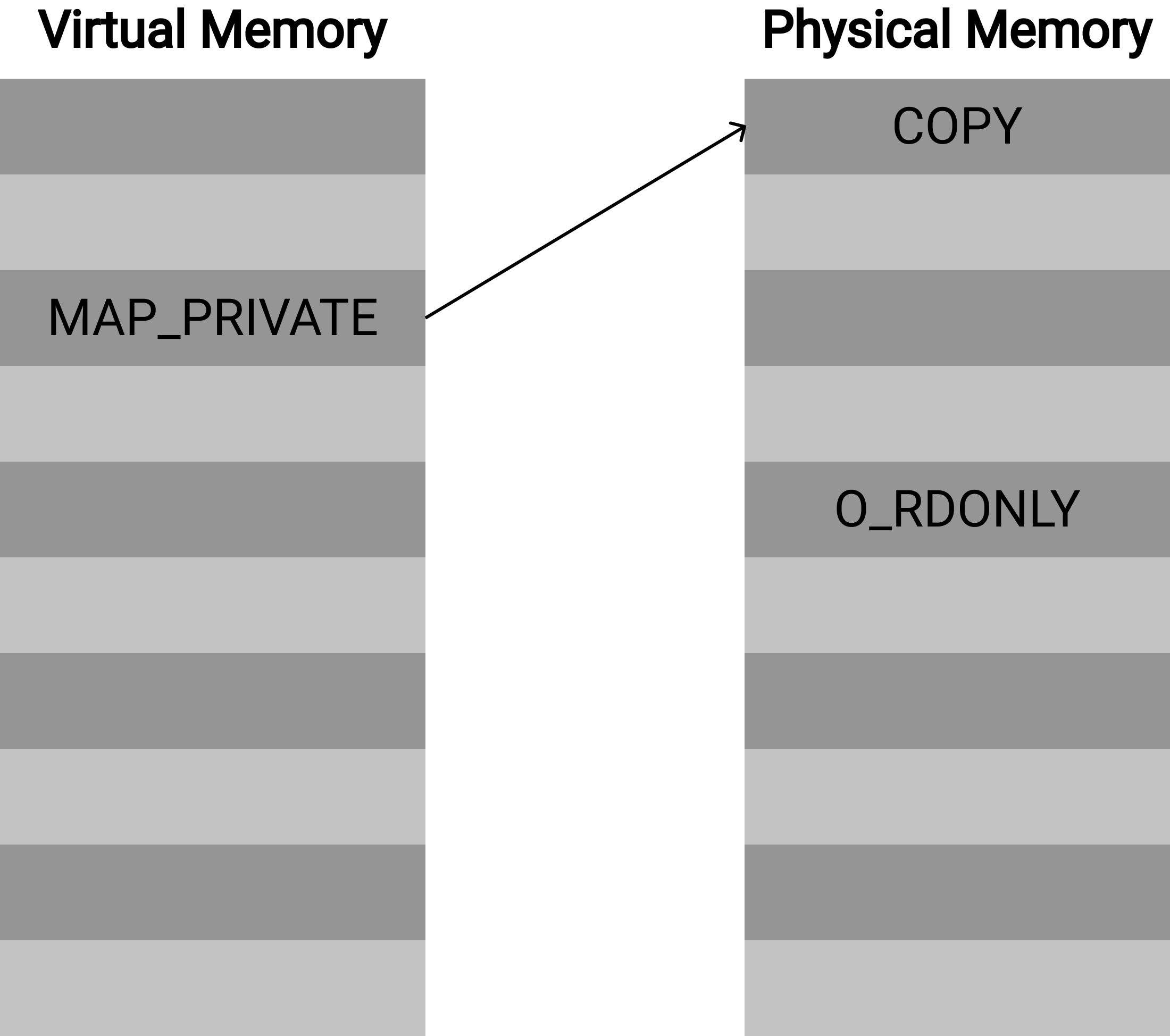
לאחר מכן נוצר מיפוי לאותו הקובץ עם הדגל MAP\_PRIVATE שמאפשר שימוש בCOW.



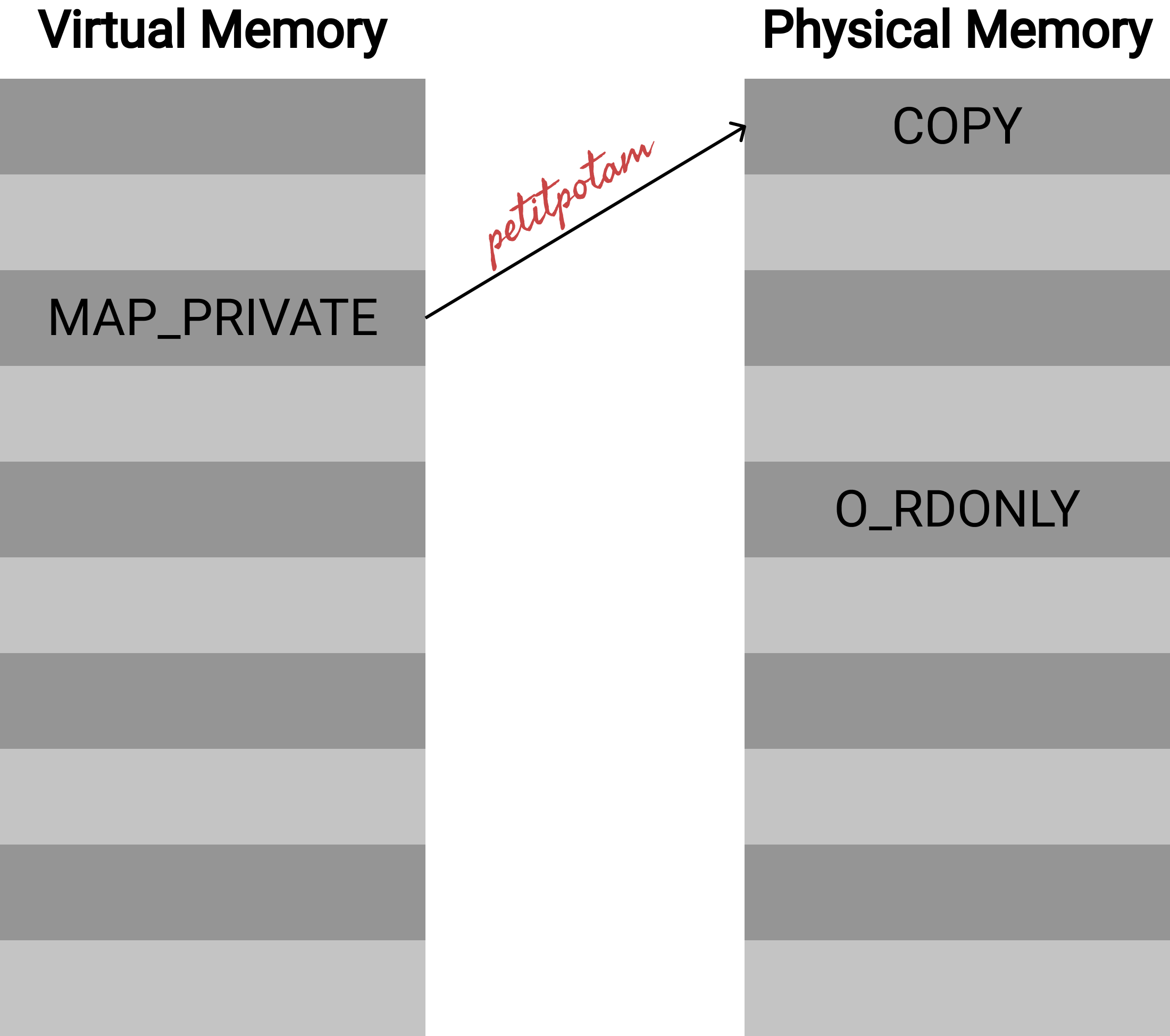
ואז מורצים שני thread-ים במקביל. אחד מבצע כתיבה למיפוי, והשני קורא לsyscall – madvice שמשנה חזרה את המיפוי מהעותק של הקובץ בזיכרון חזרה אל המקור.

התהליך של כתיבת תוכן לקובץ ממופה מחולק לשני חלקים נפרדים.

החלק הראשון הוא יצירת העותק ושינוי המיפוי אל העותק הפרטי החדש.



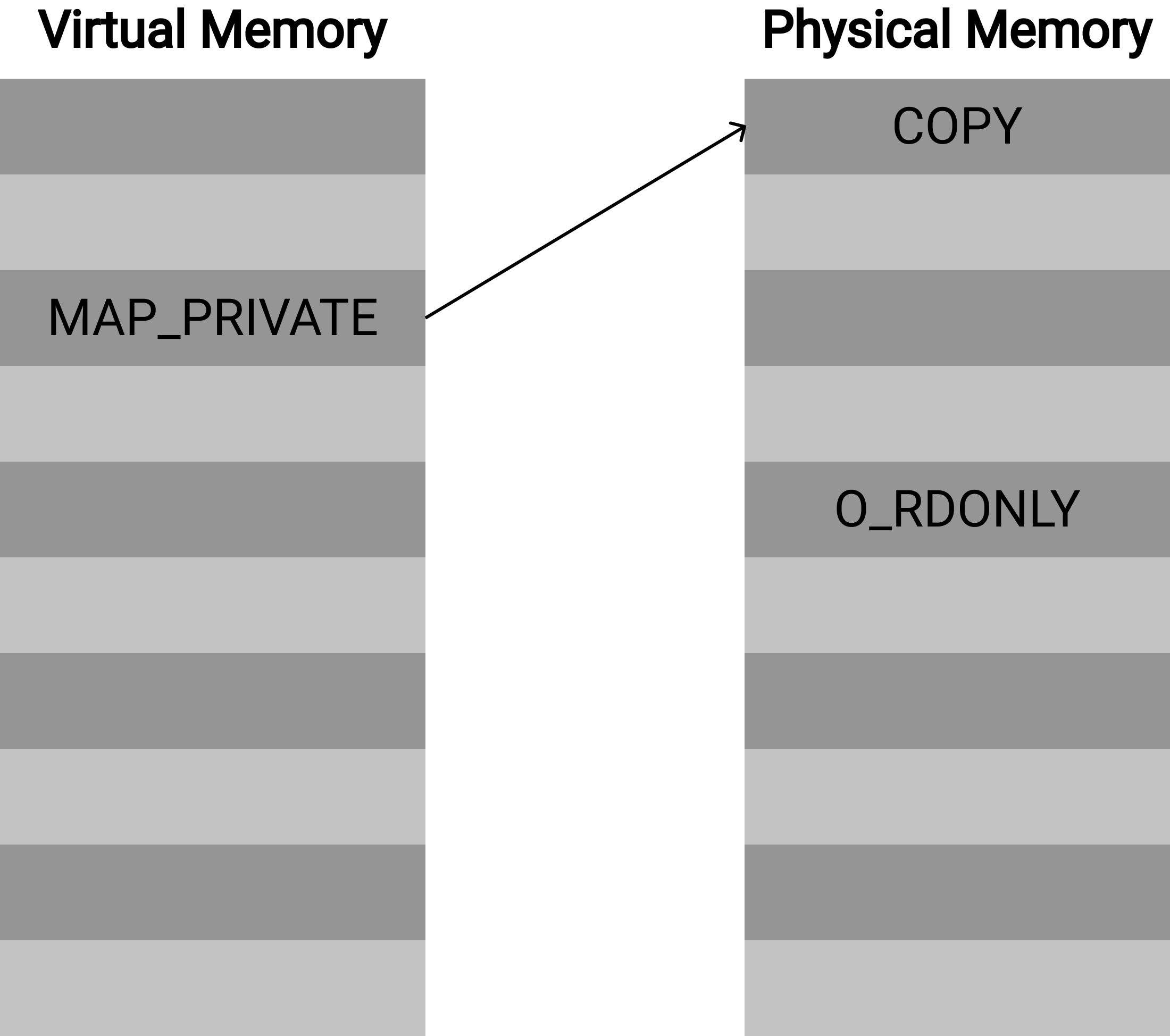
והחלק השני מבצע את הכתיבה לאותו הכתובת בזיכרון הווירטואלי.



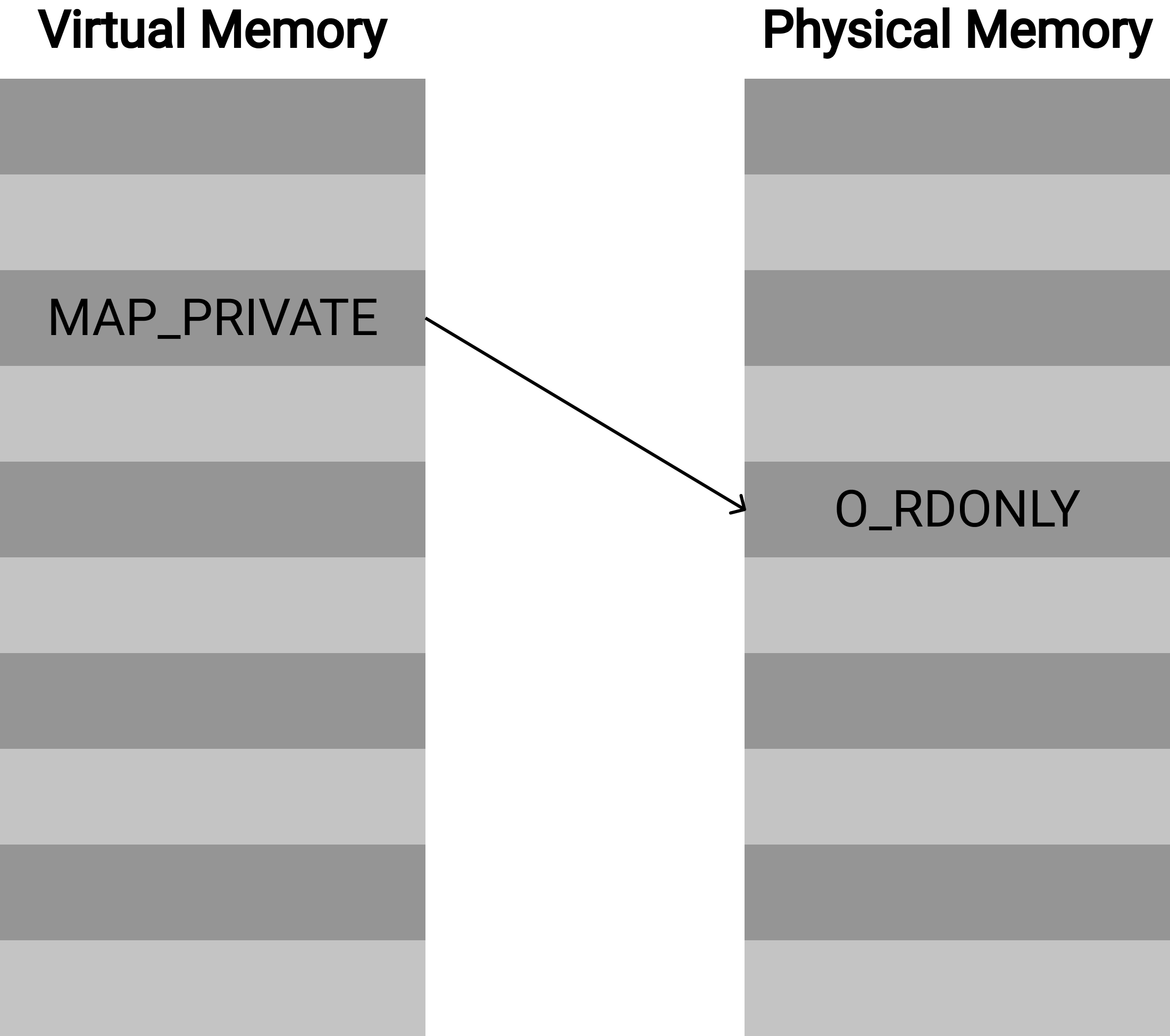
עכשיו, מישהו יכול לומר לי מהו הrace condition?

**מישהו עונה**

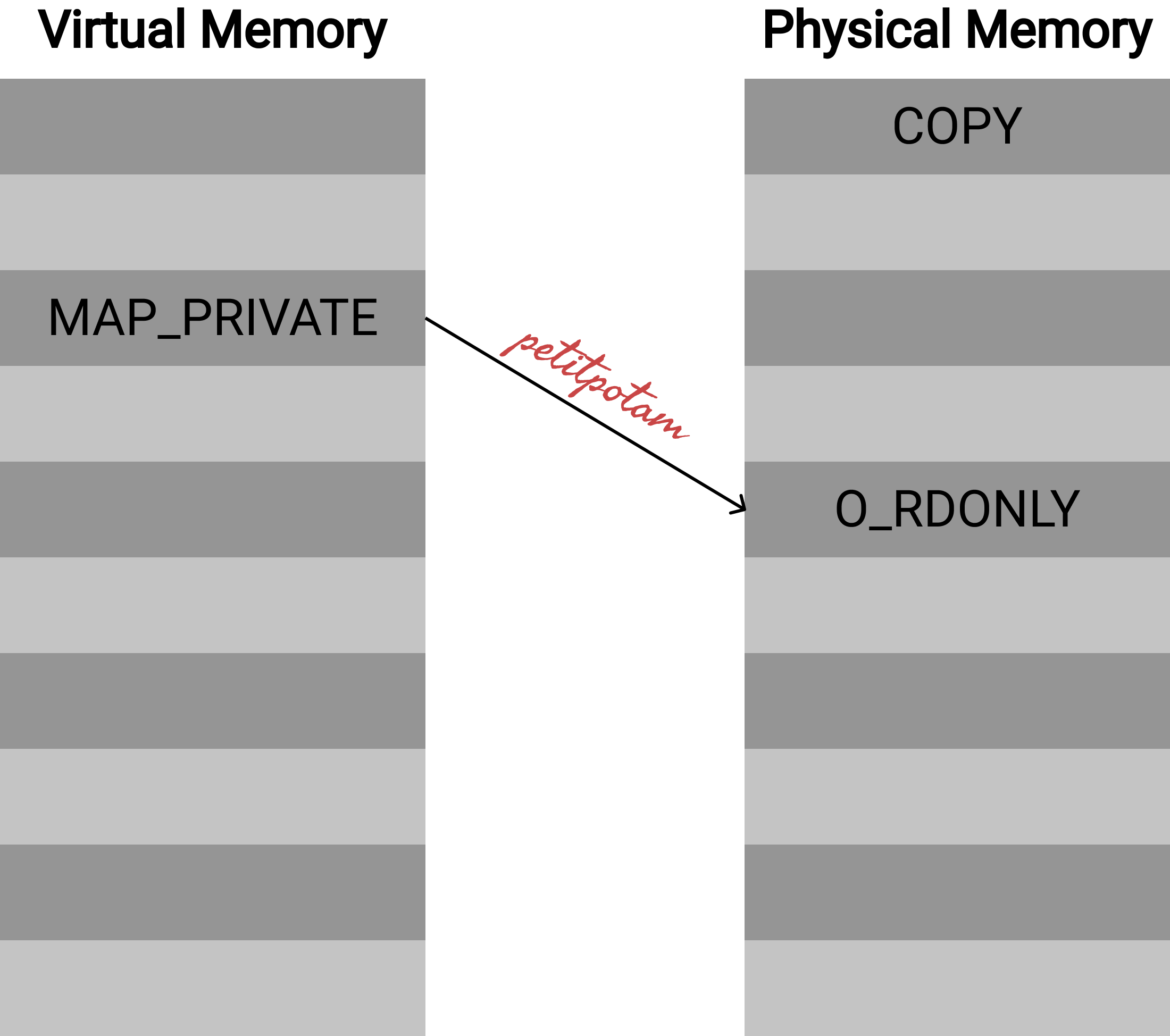
נכון. כמו שאמרתי, בין שני החלקים של הכתיבה יכול להיכנס thread אחר. החולשה מסתמכת על כך שיקרה החלק הראשון – ייווצר העותק הפרטי בזיכרון וישתנה המיפוי אליו.



ואז יקרה Context switch ויורץ madvice, מה שישנה את המיפוי חזרה אל הקובץ המקורי.



אחר כך יקרה שוב context switch, לthread של הכתיבה, ובגלל שלא מתבצעת שום בדיקה, הכתיבה תתבצע על הקובץ המקורי.



## מקורות

<https://www.youtube.com/watch?v=PCKhmPTDurg>

<https://www.digitalwhisper.co.il/files/Zines/0x4D/DW77-1-RaceCondition.pdf>

<https://man7.org/linux/man-pages/man2/madvise.2.html>

<https://www.youtube.com/watch?v=CQcgz43MEZg>

<https://www.cs.toronto.edu/~arnold/427/18s/427_18S/indepth/dirty-cow/index.html>