תיעוד החלק הרטוב – תרגיל בית מספר 2- מערכות הפעלה

מבנה הנתונים התומך במדיניות החדשה:

על מנת לתמוך במדיניות זימון התהליכים החדשה ביצענו מספר שינויים עבור מבנה ה-runqueue. הוספנו בעצם שדה נוסף מטיפוס prio\_array\_t המיועד לתהליכי SHORT וכן הוספנו שדה נוסף דומה עבור תהליכי SHORT שהם overdue.

ישנו קובץ המעטפת עבור קריאות המערכת החדשות

שינויים שביצענו בקובץ sched.h:

1. הגדרנו את המדיניות החדשה כמאקרו בדומה למדיניויות אחרות: #define SCHED\_SHORT 4
2. בתוך מבנה ה-sched\_param הוספנו את השדות הדרושים בהתאם ל-PDF התרגיל.
3. הוספנו את השדות הבאים למבנה ה-task\_struct:
   1. שדה ה-requested\_time
   2. שדה ה-number\_of\_trials
   3. שדה של number\_of\_trials\_used (שלושת השדות האחרונים תואמים להגדרת התרגיל כאשר שדה ה-used מסמן כמובן את מספר הנסיונות שכבר השתמשו בהם).
   4. שדה המתפקד כדגל שנקרא is\_SHORT\_OVERDUE וערכו שווה ל-1 במידה וזהו אכן תהליך SHORTֹ מסוג OVERDUE, אחרת יחזיק את הערך 0.
   5. שדה שנקרא reason\_CS שמחזיק עבור תהליך את הסיבה שבעטיה בוצעה איתו החלפת הקשר.

שינויים שביצענו בקובץ sched.c:

1. הוספו כלל קריאות המערכת שנדרשנו להוסיף במסגרת התרגיל
2. הוספנו את מבנה ה-switch\_info כפי שנדרש ממנו במסגרת התרגיל עבור חלק ה-statistics
3. מימשנו עבור חלק ה-statistics את הפונקציה add\_to\_statistics\_buffer אשר מוסיפה טיפוס switch\_info למבנה ה-buffer המעגלי אשר מחזיק את נתוני החלפות ההקשר שבוצעו. בנוסף הגדרנו בקובץ זה את ה-buffer עצמו לשמירת הסטטיסטיקות.
4. עדכנו את הפונקציה activate\_task כך שהפונקציה תדע להתמודד עם המקרים החדשים של תהליכים קצרים ותהליכים קצרים שפג תוקפם ויבצעו הכנסה והוצאה למבנה המתאים בהתאם למבנים שהגדרנו.
5. עדכנו את הפונקציה scheduler\_tick כך שנבצע את הורדת ה-time\_slice והחישוב המתאים כפי שתואר בתרגיל עד אשר יגמר ה-time slice ואז נטפל בהתאם.
6. עדכנו את הפונקציה schedule
7. עדכנו את הפונקציה setscheduler כך שתדע להתמודד עם המדיניות החדשה שהגדרנו. כידוע, פונקציה זו אחראית על שינוי המדיניות של התהליך ולכן היינו צריך לטפל בכלל המקרים ולדאוג שרק שינויי מדיניות תקינים בהתאם לדרישת התרגיל ובאופן כללי יוכלו להתבצע. אנו מוודאים במסגרת פונקציה זו את תקינות הפרמטרים, אנו דואגים להוציא את התהליך מהמבנה בו נמצא ולדאוג שיגיע למבנה אליו אמור להגיע במסגרת המדיניות החדשה ואנו מדליקים את דגל ה-need\_resched.
8. עדכנו את הפונקציה try\_to\_wake\_up עברנו על כלל המקרים שייתכנו של הערת תהליך בהתאם למדיניויות השונות ואם אכן קיימת עדיפות לתהליך הנוכחי שרץ במערכת וזאת בהתאם לדרישות העדיפות המתוארות בתרגיל.

שינויים שביצענו בקובץ fork.c:

בקבצים syscalls.h, unistd.h ו-entry.S:

עדכנו את כלל ההגדרות הנדרשות על מנת שקריאות המערכת אשר הוספנו יעבדו.

הערות לגבי חלק ה-statisticcs:

כפי שכבר צוין, דאגו להגדיר במסגרת הקובץ sched.c את המבנה struct switch\_info וכן את ה-buffer המעגלי. מדובר ב-BUFFER שהוא בעצם מערך בגודל 150 כאשר כל איבר בו מסוג struct switch\_info. את שדה ה-reason ישנם מספר מקומות בהם היינו צריכים לעדכן את ערכו בהתאם למספר הסיבה:

1. בפונקציה do\_fork כאשר נוצר תהליך חדש וגם בפונקציה setscheduler
2. בפונקציה do\_exit
3. בפונקציה sched\_yield
4. יוצא ל-waiting
5. בפונקציה try\_to\_wake\_up
6. בפונקציה scheduler\_tick כאשר ה-time slice נגמר.

כל הקבצים הרלוונטים כפי שתוארו לעיל גם עודכנו על מנת לתמוך בקריאת המערכת החדשה שהוגדרה והיא sys\_get\_scheduling\_statistic