

ממן 13 :

2.א) **היתרון** של מדיניות זו הוא טיפול מהיר בבקשות שנכנסו לאחרונה. מהנתון, הבקשה הראשונה שנכנסת היא הבקשה שהגיעה לאחרונה, מדיניות זו תוביל לזמן המתנה נמוך לקריאות שנכנסו לאחרונה. ב) **החסרון** של מדיניות זו הוא שהמדיניות עלולה לגרום להרעבה של תהליכים שדורשים שימוש בדיסק. במקרה בו קיימים תהליכים ששולחים בקשות לדיסק, עלול להיווצר מצב בו הקריאות המטופלות הן הקריאות החדשות שהגיעו כלומר הקריאות הישנות ימתינו המון זמן לטיפול.

3. המדיניות בה הייתי משתמשת: i-node

הסבר:

מדיניות זו היא שיטה לאחסון קבצים בה לכל קובץ מוקצה i-node המכיל מידע אודות הקובץ, למשל גודל הקובץ, מיקום בדיסק והרשאות

מדיניות זו מותאמת היטב למערכות בעלות טווח רחב של גדלי קבצים.

אחד מיתרונות מדיניות זו היא שהיא מאפשרת לקובץ לתפוס מקום לא רציף בדיסק, כך ניתן לאחסן קבצים קטנים ביעילות משום שלא חייב למצוא בלוק גדול ורציף לאחסון הקובץ בדיסק.

בנוסף במדיניות זו מאפשרת גמישות בהקצאת מקום בדיסק. היא מאפשרת למערכת להקצות מקום במקומות שונים ברחבי הדיסק לשמירת הקובץ וכך בעצם לאפשר שימוש טוב יותר בדיסק ופחות ריסוק.

4.א) דמיון:

שני המודלים שואפים לשיפור הבידוד והאבטחה על ידי יצירת סביבות מבודדות עבור אפליקציות שרצות. בשני המודלים, מתבצעת יצירה של namespace נפרד עבור כל אפליקציה:

- באנדרואיד מתבצע על ידי יצירת user IDs
- ב xv6 מתבצע על ידי PID namespace

פעולות אלו מגבילות את המשאבים וההרשאות של כל אפליקציה למערכת ההפעלה.

שוני:

ההבדל המרכזי בין המודלים הללו הוא המימוש עצמו (כלומר user IDs VS. PID namespaces)

PID namespaces יוצרים סביבה מבודדת לכל אפליקציה בעוד שבאנדרואיד user ID's

מוודאים שאין לאפליקציה הרצה גישה למידע רגיש שלא אושר על לשימוש על ידי המשתמש.

ב) הבידוד בא לידי ביטוי על ידי יצירת user ID's ייחודי עבור כל אפליקציה המתוקנת במכשיר.

כאשר אפליקציה מותקנת במכשיר המשתמש, מוקצה לה ID ייחודי וכל קבצי האפליקציה מאוכסנת ב directory המשווייך ל ID. הגישה ל directory זו מתבצעת רק דרך האפליקציה ולא ניתן לגשת ל directory זה מאפליקציה אחרת לדרך המשתמש.

כך בעצם האפליקציה יכולה לגשת לערוך רק קבצים שרלוונטים אליה וכך בעצם מתבצע הבידוד.

ג) היתרון של ה COW שמערכת אנדרואיד משתמשת בה למימוש מודל התהליכים שלה היא יתרון הבטיחות המוגברת – COW מאפשרת בידוד טוב יותר בין תהליכים באמצעות מניעת עדכון ה memory pages של תהליך אחד על ידי תהליך אחר.

5. תיאור שתי השיטות:

1. הדרך הראשונה היא לשמור capability list במבנים פנימיים שנמצאים במרחב זכרון ששייך למערכת ההפעלה ואז כל שינוי של הרשימה בהכרח יעבור דרך המערכת הפעלה.

2. הדרך השנייה דומה לדרך הראשונה , כלומר שומרים capability list באותה דרך אבל אז מצפינים אותה כך שרק מערכת ההפעלה תוכל לפענח את הרשימות ולבצע שינוי של הרשימות.