

שאלה 1 :

הם נשמרים בצורה זו בשביל שהפרמטר הראשון יהיה בראש של המחסנית ובדרך זו מתאפשר ל variadic פונקציות לעבוד בצורה טובה (printf במשתנה הראשון יש את מספר של הפרמטרים כאשר הוא בראש של המחסנית קל יותר לגשת אליו ולהפעיל printf בצורה טובה).

שאלה 2)

במחשב יש שתי מחסניות, הראשונה נמצאת במצב משתמש (user mode) והשניה נמצאת בגרעין (kernel mode) אז כשאנחנו קוראים לקריאת מערכת כלשהי, אנחנו נעבור מ user mode ל kernel mode ואז המחסנית מתחלפת למחסנית הגרעין (כותבים למחסנית גרעין במקום מחסנית משתמש).

שאלה 3)

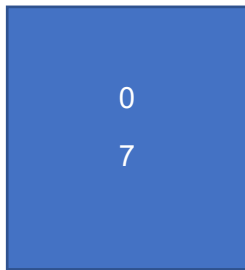
Output: 8 or 813 or 183 or 138 or(nothing)

מקרה 1) (fork() נכשלת, נכנסים ל if הראשון, ואז התוכנית לא מדפיסה שום דבר
מקרה 2) (fork() הצליחה וגם execv הצליחה אז בעצם פו יש שני תהליכים שרצים באותו זמן התהליך הראשון, תהליך האב (שנכנס ל (else if (pid > 0) התהליך השני, תהליך הבן (שנכנס ל else)
אבל לא משנה הפלט בכל המקרים יהיה זהה כי תהליך הבן לא מדפיס שום דבר והפלט יהיה 8 (ה id של האב)
מקרה 3) (fork() הצליחה אבל execv נכשלה אז בעצם פו יש שני תהליכים שרצים באותו זמן התהליך הראשון, תהליך האב (שנכנס ל (else if (pid > 0) התהליך השני, תהליך הבן (שנכנס ל else) אז פו הפלט יהיה קצת בעייתי כי בכל רגע לא יודעים מי מדפיס ראשון (האב או הבן) אז הפלט יהיה 138 או 183 או 813.

שאלה 4 :

יש לנו כמה מקרים :

(1) אם fork הצליח תהליך האב נכנס ל wait ומחקה עד שהבן שלו יסיים תהליך הבן לא נכנס ל if כי ערך המוחזר מהפורק הוא 0 הוא מדפיס את ערך value שהוא 0 מוסיפים 4 לו ו שולחים exit עם status=4 מסיימים ה wait מזהה שאחד מתהליכי הבן הסתיימו חוזרת ותהליך האב ממשיך מעדכנים ש value תהיה שווה ל 4 שהוא הערך שהוחזר לתוך status ונקח אותו באמצעות wexitstatus מוסיפים 3 value=7 מדפיסים :



(2) אם ה fork נכשלה מחזירה 1- נכנסים ל if בתהליך הנוכחי עושים wait אבל אין לנו בנים חוזרים מיד value=0 כי לא שינינו אותה לכן wexitstatus תחזיר גם 0 נוסף 3, value=3 נדפיס ונעשה exit נסיים .

