Important SOLARIS / UNIX System Calls

System Call	Description	Notes	Examples

fork	נקראת מתוך תהליך שרץ. משכפלת את התהליך הנוכחי שרץ, וממשיכה את הריצה של המקורי(האב) והמשוכפל (הבן). האב מקבל כערך החזרה את הpid של הבן, והבן מקבל ערך 0. ההבדלים בין האב לבן הם בpid. אם התהליך מכיל תרדים, אזי זה תלוי במערכת ההפעלה אם תשכפל אותם או לא.	אם אין מקום לתהליך בטבלאת התהליכים, ערך ההחזרה הוא 1-, ורק התהליך המקורי(האב) ממשיך לרוץ. לרוב אנו משתמשים בפקודה זו על מנת להריץ קובץ הרצה אחר מהבן, ע"י שימוש בפקודות exec למיניהן.	int pid = fork();
			if(pid<0) { //fork failed code}
			if(pid!=0) { // father code }
			else { //child code }

execvp	טוענת קובץ הרצה ומבצעת אותו במקום התהליך הנוכחי. נשים לב, שהיא דורסת את הcontext של התהליך שביצע הפקודה, אך עובדת תחת אותו pid. נשים לב, שהפקודה גם מאפסת את טבלת הסיגנלים של התהליך הקורא.	אם השימוש בפקודה לא חוקי או לא ניתן לבצעה מסיבה כלשהיא, חוזר ערך 1- והתהליך הקורא ממשיך בפעולתו. לרוב משתמשים בexecvp אחרי fork.	char** params = {"dirName"};
			int res = execvp("md", params);
			if(res < 0) { //execvp failed }
			// else, running md code

wait	צעד ראשון בדרך לסינכרון ברמת התהליכים. ברגע שישנו אב ובן לאחר ביצוע fork, אם האב מבצע wait, הוא מחכה שהבן יגמור. אם לאבא יש הרבה בנים, אזי הפקודה תסתיים שאחד הבנים יחזור.	ניתן גם לספק לפקודה משתנים נוספים שיתמלאו ברגע שהפקודה תסתיים, ויתנו לנו מידע כמו באיזה מצב הבן הסתיים, איזה בן הסתיים וכולי. אם כשהאב קורה לפקודה ובניו כבר סיימו (או שאין לו כלל בנים), הפקודה מיד תחזור והאב ימשיך בפעולתו	// the process creates a child and
			// waits for him to finish
			if(fork()!=0) { wait(); }
			else { // child code }

****** Signal Table***	לכל תהליך יש טבלת סיגנלים שבה השורה ה i מייצגת את הinterrupt מספר i, ומכילה ביט ignore, ומצביע לפונקציה. ללא כל התערבות, או שבביט הignore יהיה true או false ויהיה מצביע לפונקציה של מערכת ההפעלה שעושה דברים כגון suspend, resume, stop, interrupt (=stop).	ברגע ששולחים תהליך סיגנל עם פקודה עם ערך הסיגנל שלילי – אנחנו שולחים את הסיגנל לכל קבוצת התהליכים שנמצאת בforeground (כלומר לכל התהליכים עם אותו group id), ואם אנו שולחים עם ערך סיגנל חיובי אנו שולחים רק להתהליך הספיציפי שאליו נשלחת ההודעה. טריק נפוץ – בין הfork לexecvp להוסיף disable על סיגנלים. ניתן להחליף לתהליך את הgroup id, ובכך אם הקבוצה הקודמת שלו מקבלת הודעה, היא לא מגיעה אליו.	Kill –value processID // for group Kill value processID // only to pid
pause	פקודה העוצרת את התהליך שרץ וגורמת לו לחכות עד שיקבל signal (בהנחה שהסיגנל שקיבל איננו במצב ignore – אחרת הוא איננו יוצא מpause). ברגע שהוא מקבל סיגנל הוא יוצא מpause, מטפל בסיגנל, וממשיך בריצתו הסדירה.		

עריכה: שאול קובל ® מערכות מחשבים - כל הזכויות שמורות © אסור לשכפול - http://www.coval.net