OS

7 באפריל 2025

תוכן העניינים

0.1 Introduction

0.2 יצירת תהליך וסיום

- תהליך אחד (ההוֹרה) יכול ליצור אחר (הילד).
- יש יצירה של PCB חדש שמוקצה ומאותחל.
- שיעורי בית: הרץ את הפקודה 'ps auxwww' ב-shell; PPID הוא PID של ההורה.

0.3 ב-POSIX, התהליך הילד מיירש את רוב מאפייני ההורה

- UID, קבצים פתוחים (צריך לסגור אם אין צורך; למה?), cwd, וכו'.

0.4 בעת הביצוע, PCB מתנהל בין תורים שונים

- לפי גרף שינוי המצב.
- תורים: runnable, i event for sleep/wait (i=1,2,3…).

0.5 לאחר שתהליך מת (exit()s / interrupted), הוא הופך לזומבי

- ההורה משתמש בקריאת מערכת *wait כדי לנקות את הזומבי מהמערכת (למה?).
- משפחת קריאות המערכת: wait, wait, waitpid, waitid, wait3; wait4 לדוגמה:

0.6 ההורה יכול לישון/לחכות לילד שלו לסיים או לרוץ במקביל

- *wait() יחסום אלא אם כן ניתן WNOHANG ב-options.
- שיעורי בית: קרא את 'man 2 wait'.

```
([]argv* char ,argc int)main int
}
;()fork = pid int
} ( 0==pid )if
//
child //
//
,n"\d%=son d%=parent)"printf
;(()getpid ,()getppid
{
} ( 0 < pid )if else
//
parent //
//
,n"\d%=son d%=parent)"printf
;(pid ,()getpid
{
associated string print // } else
errno with //
;(failed" ()fork)"perror
{
;0 return
•{
-PCB new a initializes ()fork
-value parents' on Based
•queue runnable to added PCB
-processes 2 are there Now
•point execution same At
-space address new Childs'
parents' of copy Complete
•difference… one with ,space
-twice returns ()fork
-0<pid with ,parent the At
•0=pid with ,child the At
•?order printing the Whats'
-variable global a - errno'`
syscall last of num error Holds
signals & processes - (234123) OS
```