**ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΉΜΑΤΑ 20/10**

**Άσκηση 1**

Δίνεται ένα σύστημα Linux που χρησιμοποιεί τον χρονοδρομολογητή Ο(1) και 4 διεργασίες Α,Β,Γ,Δ με αριθμούς προτεραιοτήτων 107, 105, 112 και 114 αντίστοιχα. Οι διεργασίες κατέφθασαν σε χρόνο t=0. Να βρείτε τα κβάντα που θα πάρουν μετά την πρώτη εκτέλεση και να δείξετε ην κατάσταση των ουρών active/expired. Ο τρόπος υπολογισμού των κβάντων ακολουθεί την πολιτική του Ο(1) χρονοδρομολογητή και οι διεργασίες είναι σε κατάσταση sleep κατά 50%, 25%,75%, και 50%, αντίστοιχα, κάθε φορά που εκτελούνται.

1. Στο ενδιάμεσο, θεωρήστε ότι δεν εισέρχονται νέες διεργασίες στο σύστημα
2. Θεωρήστε ότι εισέρχονται 2 διεργασίες με προτεραιότητα 103 και 120, οι οποίες περιμένουν τον επόμενο κύκλο εκτέλεσης.

\*Στον 1ο κύκλο εκτέλεσης, όλες τρέχουν για 1s

Απάντηση

Κύκλος εκτέλεσης: το ΛΣ διαβάζει της Διεργασίες που είναι σε κατάσταση Ready σε όλες τις ουρές και τις τρέχει ανεξάρτητα με το αν ενδιάμεσα μπουν άλλες διεργασίες.  **O(1)**

1. Διαβάζουμε το Bitmap

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 100 |  | 105 |  | 107 |  | 112 |  | 114 |  | 130 |
|  |  |  | 1 |  | 1 |  | 1 |  | 1 |  |  |

1. Static Priority -> SPA=107, SPB=105, SPC=112, SPD=114

Default priority =120

1. SPB=105. Interactive το περισσότερο χρόνο εκτέλεσης κάνουν διερασίες Ι/Ο. Άρα αυτές οι διεργασίες δεν κουράζουν την CPU. Όσο καλύτερο ποσοστό του χρόνου γίνεται Ι/Ο, η διεργασία παίρνει Bonus. Μεγ.Bonus=>Χαμ.DP (Dynamic Priority) =>περισσότερα κβάντα.

-Non-Interactive: αποστέλει πολό την CPU,χάνει σε bonus άρα το DP της αυξάνεται (λιγότερα κβάντα )

* Για τις τιμές Bonus 0-4, το DP αυξάνεται αρά χάνουμε P

Πίνακας-eclass

* Για τις τιμές Bonus=5, ίδια P
* Για τις τιμές Bonus>5, το DP μειώνεται άρα κερδίζουμε P

-> SPA=107, SPB=105, SPC=112, SPD=114

* Τρέχει η Β για 1 sec, κάνει Ι/Ο για 25% του χρόνου, δηλαδή 250 ms. Άρα Bonus= 2. Άρα DPΒ= MAX[ 100, { min (**105-**2+5), 139 } ] = MAX[ 100, { min (**108**), 139 } ] = 108. Άρα η διεργασία θα μεταφερθεί στην ουρά 108 (έχασε προτεραιότητα)
* Αντίστοιχα η Α, με 50% του χρόνου, 500 ms ->bonus=5. DPA= MAX[ 100, { min (**107**), 139 } ] = 107
* Αντίστοιχα η C, με 75% του χρόνου, 750 ms ->bonus=7. DPA= MAX[ 100, { min (**110**), 139 } ] = 110
* Αν Pr<120 =>(140-PR)\*20ms
* Αν Pr>=120 =>(140-PR)\*5ms

QA=(140-QA)\*20=(140-107)\*20=660ms

QB=(140-QB)\*20=(140-108)\*20=640ms

QC=(140-QC)\*20=(140-110)\*20=600ms

QD=(140-QD)\*20=(140-114)\*20=520ms

QE=(140-QE)\*20=(140-139)\*20=5ms

-Έστω μια διεργασία Ε, η οποία μετά τον 1ο κύκλο είχε DPE=139

1. Ο Ο(1) δημιουργεί μεγάλη ανισορροπία στους χρόνους εκτέλεσης των διεργασιών.

|  |  |  |
| --- | --- | --- |
|  |  | expired |
| 105 |  |  |
|  |  |  |
| 107 | A |  |
| 108 | B |  |
|  |  |  |
| 110 | C |  |
|  |  |  |
| 112 |  |  |
|  |  |  |
| 114 | D |  |

|  |  |  |
| --- | --- | --- |
|  |  | active |
| 105 | B |  |
|  |  |  |
| 107 | A |  |
|  |  |  |
| 112 | C |  |
|  |  |  |
| 114 | D |  |

-Διάβασμα bitmap

ACTIVE->δομη περιέχει τις ουρές διεργασία που είναι Ready. Όταν μία διεργασία ολοκληρωθεί (τέλος q) περνάει στην δομή EXPIRED στην νέα της ουρά

1. Εκτέλεση
2. Υπολογισμός νέων DP
3. Υπολογισμός νέων q

Η Β τρέχει, ολοκληρώνεται και το ΛΣ μεταφέρει την διεργασία σε σταθερό χρόνο στην ουρά 108 της δομής EXPIRED. Το ίδιο γίνεται και με τις άλλες. Δεν χάνουμε χρόνο. Στην συνέχεια, πρέπει οι διεργασίες να γυρίσουν στην ACTIVE για τον επόμενο κύκλο. Αυτή η μεταφορά δεν είναι Ο(1). Είναι Ο(Ν) και εξαρτάται από το πλήθος Ν των διεργασιών.

BITMAP

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 100 |  | 105 |  | 107 |  | 112 |  | 114 |  | 130 |
|  |  |  | 1 |  | 1 |  | 1 |  | 1 |  |  |

Επαναλαμβάνεται ο κύκλος π.χ. η Α θα τρέξει για 660ms αν υποθέσουμε ότι bonusA=5, τότε θα μεταβεί στην ACTIVE στην ουρά 107. Για την Α, το bonusA=5 σημένει από 50-59% των 660ms, ξοδεύεται σε Ι/Ο.

|  |  |  |
| --- | --- | --- |
|  |  | expired |
| 103 | Ε |  |
| 105 |  |  |
| 107 | A |  |
| 108 | Β | active |

|  |  |  |
| --- | --- | --- |
| 110 | C |  |
| 112 |  |  |
| 114 | D |  |
| 120 | F |  |

B=ΣΕΙΡΑ ΕΚΤΕΛΕΣΗΣ

E=1 sec

A=660 ms

B=640 ms

C=600 ms

D=520 ms

F=1 sec

Έστω ότιτο ΛΣ περιμένειτον επόμενο κύκλο και οτι οι E, F θποβάλλονται καθώς τρέχει η Β

Εξετάζοντας το BITMAP,μόλις η Β πάει στην EXPIRED, θα τρέχει η E τελικά, η σειρά εκτέλεσεις θα είναι B,E,A,C,D,F

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 103 |  | 105 |  | 120 |
| 1 |  | 1 |  | 1 |

Συμπέρασμα**=>+\***ΤΑΧΥΤΑΤΑ\*Μεταξύ διεργασιών με ίδια προτεραιότητα είναι αποδοτικός**.-.** Όταν υπάρχει μεγάλη οντότητα στις προτεραιότητες, υπάρχει τεράστια διαφορά στα q

π.χ. DP=139, q=3

DP=100, q=800

* Δεν είναι δίκαιος
* Διεργασία με μεγάλη τιμή DP (χαμηλή προτεραιότητα ) μπορεί να αργήσουν πολύ να τρέξουν