Project 3
Λειτουργικά Συστήματα

Ονοματεπώνυμο: Γεωργόπουλος Ιωάννης
Α.Μ.: 1115201800026

Δομές Δεδομένων:

sharedMemory: Περιέχει γενικά τις δηλώσεις συναρτήσεων και μεταβλητών που ειναι σημαντικές για την εκτέλεση των λειτουργιών μετάδοσης των μηνυμάτων. Ειδικότερα περιέχει τις δηλωσεις συναρτήσεων που χρησιμοποιούνται για την διαχείριση της κοινής μνήμης, τα ονόματα των σημαιοφόρων που χρησιμοποιούνται, το μέγεθος των μπλοκ κοινής μνήμης, τα ονόματα των μπλοκ κοινής μνήμης.
Συγκεκριμένα:
get_shared_block():
Παίρνει το key από το filename με τη βοήθεια της ftok(), το οποίο (filename) πρέπει να αντιστοιχεί σε πραγματικό αρχείο μέσα στον φάκελο των αρχείων. Μετά με την shmget() παίρνει ένα id το οποίο είναι χαρακτηριστικό για το συγκεκριμένο key.
attach_memory_block(): Παίρνει το id του μπλοκ κοινής μνήμης με τη βοήθεια της get_shared_memory() και του filename και το χρησιμοποιεί στην συνάρτηση shmat() για να κανει attach σε ένα δείκτη την διεύθυνση του μπλοκ κοινής μνήμης και επιστρεφει αυτή την τιμη.
detach_memory_block():
Παιρνει τον pointer που δείχνει σε μπλοκ κοινής μνήμης και το κάνει detach από τον pointer στον οποίο είχε ανατεθεί προηγουμένως.
destroy_memory_block():
Παίρνει το ονομα ενος μπλοκ κοινής μνήμης και καταστρέφει το μπλοκ με το συγκεκριμένο όνομα.

linkedList:
Αυτή η λίστα χρησιμοποιείται αρχικά από τον generator για να μπορεί να παρακολουθεί τις διεργασίες που τρέχουν εκείνη τη στιγμή για να μπορεί να στείλει σε σωστό χρόνο το μηνυμα τερματισμου τους στον memory manager.
Επίσης χρησιμοποιείται από τον memory manager για την αποθήκευση όλων των διεργασιών του simulation ώστε στο τέλος να μπορέσει να δημιουργηθεί το logFile.

fifo:
Η fifo χρησιμοποιείται ως λίστα αναμονής διεργασιών για τον generator και κάθε δευτερόλεπτο βγαινουν διεργασίες από αυτή τη λίστα για να μπουν στη μνήμη και στη λίστα τρεχούμενων διεργασιών.

memory:
Η μνήμη υλοποιείται ως ένας πίνακας ακεραίων όπου αν έχουμε best-fit η worst-fit όταν ένα κελί έχει τιμή -1 είναι κενό και όταν έχει την τιμή ενός process id σημαίνει ότι εκεί βρίσκεται ένα κομμάτι μιας διεργασίας. Αν έχουμε buddy τότε το κενο κελι μνήμης αντιπροσωπεύεται με 0 και εκεί που βρίσκεται η διεργασία τα κελιά έχουν την τιμή του process id της διεργασίας

και επίσης τα κομματια μνήμης που δεν μπορεί να μπει άλλη διεργασία λογω του αλγόριθμου του buddy αλλα δεν καταλαμβανονται άμμεσα από μια διεργασία αντιπροσωπεύονται με την αρνητική τιμή του process id της διεργασίας που κάνει occupy το συγκεκριμενο segment μνήμης.Επίσης κάνω τη παραδοχή ότι το μέγεθος της μνήμης είναι της μορφής S=2^n με n μη αρνητικός ακέραιος.Στο worst-fit και best-fit για την εισαγωγή μιας διεργασίας βρίσκω όλα τα κενά της μνήμης και αντιστοιχα βάζω την διεργασία στο χειρότερο η στο καλύτερο κενό ανάλογα με το αν είναι worst-fit η best-fit. Στο buddy αρχίζω και χωρίζω την μνήμη δια δυο σε κομματια ισα μεχρι να βρω τα κομματια που ειναι λιγο μεγαλυτερα η ισα του μεγεθους της διεργασίας και έπειτα δοκιμαζω να την βάλω σε όποιο από αυτα χωραέι πηγαίνοντας από το αριστερό προς το δεξιά κομματι μνήμης. Όταν βρω ενα κομματι που να μην το απασχολεί άλλη εργασία αρχιζω και βαζω την διεργασία στο κομμάτι αυτό απο αριστερα προς τα δεξιά. Αν υπαρχει υπολειπόμενος χώρος στο κομματι που μπηκε θα το γεμίσω με την αρνητική τιμή του process id της διεργασίας που το χρησιμοποιεί. Αυτή η διαδικασία γίνεται αναδρομικά.

generator:

Αποτελεί τον γεννήτορα των εικονικών διεργασιών και ο οποίος ευθυνεται για την γεννηση αρχικοποίηση τους καθώς και την έναρξη και τον τερματισμό τους(VPStart and VPStop) καθώς και τον τερματισμό του simulation. Επίσης διαχειρίζεται την λίστα αναμονής (VPwaitingfifo) και στο τέλος του προγράμματος στέλνει στον memory_manager στοιχεία από τη λίστα αυτή για να δημιουργηθεί κατάλληλα το logFile από τον memory_manager. Επίσης έχει μια επιπλέον λίστα την (VPRunningList) οπου καταγράφει τις διεργασίες που τρέχουν εκείνη τη στιγμή αφαιρώντας καθε δευτερόλεπτο ενα second από το χρόνο λειτουργίας τους ώστε να ξέρει ποτε να στείλει το μήνυμα τερματισμού διεργασίας(VPStop).Το simulation διαχειριζεται ο generator το οποίο simulation καταγράφει τον χρόνο ανα δευτερόλεπτο και κάθε δευτερόλεπτο κάνει τα καταλληλα VPStop,VPStart και εισαγωγές και εξαγωγές διεργασιών από τη λίστα αναμονής εξαρτώμενο βέβαια κάθε φορά από το αν υπάρχει διαθέσιμος χώρος στη μνήμη.

memory_manager:

Αποτελεί τον διαχειριστή της μνήμης που παιρνεί τα καταλληλα μηνύματα από τον generator VPStart, VPStop και term.Επίσης δέχεται το σήμα nextSec ώστε να γνωρίζει σε ποιό δευτερόλεπτο της προσομοίωσης βρισκόμαστε για να μπορεί να καταγράφει τα δευτερόλεπτα εναρξης και τερματισμού των διεργασιών. Ακόμα υπολογίζει και τις μετρικές απόδοσης: το γινόμενο χρόνου-μνήμης , τη μέση τιμή και τη διακύμανση του μεγέθους των κενών μνήμης. Όταν δέχεται το μήνυμα VPStart ελέγχει αν χωράει το process στη μνήμη και το βαζει. Αν χωράει στελνει πισω οτι μπηκε στη μνήμη επιτυχως και αν δεν χωραει στέλνει πίσω οτι δεν μπορεσε να μπει στη μνήμη.

Γενική λειτουργία:

Ξεκινόντας από τον generator αρχικοποιουνται τα ορίσματα και οι δομες που θα χρησιμοποιηθούν στο simulation. Ξεκιναμε ανα δευτερόλεπτο αρχικά κοιτωντας αν πρεπει να τερματισει καποιο process και σε αυτα που πρεπει να τερματίσουν στέλνουμε το VPStop μαζί με το procId για την αναγνώριση τους. Μετά κοιταμε αν πρέπει να γεννησουμε νεα διεργασία. Αν ναι την γερναμε και την βαζουμε στο waiting list επειτα παιρνουμε το waiting list και στελνουμε VPStart για όσες VP χωρανε στη μνήμη ή μέχρι να αδειάσει η λίστα αναμονής περιμένοντας παντα να πάρουμε μήνυμα από τον memory manager για την

επιτυχη ή την μη επιτυχή εισαγωγή της VP στη μνήμη. Έπειτα μειώνουμε το χρονο εκτελεσης των τρεχούμενων VP κατα ένα second(runTimeFunction) που βρίσκονται στη VPRunningList.Και πριν πάμε στην επόμενη επανάληψη στέλνουμε στο memory manager οτι περασε ένα second ώστε να μπορεί και αυτός να κρατάει χρόνο.

logFile:

Η δημιουργία του log File είναι δουλειά της διεργασίας memory manager. Αποθηκεύει σε μια λιστα όλα τα VPs που περασαν από την μνήμη είτε τελειωσαν είτε όχι και στο τελος λαμβάνει από τον generator και τα VPs που δεν μπηκαν καθόλου στη μνήμη έτσι ώστε να γράψει στο logFile όλες τις διεργασίες που γεννηθηκαν, το processId τους σε ποιό δευτερόλεπτο ξεκινησαν()αν ξεκινησαν) σε ποιο σταματησαν(αν σταματησαν) πόση μνήμη χρησιμοποιησαν και πόσο χρόνο περιμεναν στο waiting list.
Επίσης υπολογίζει το γινομενο χρόνου μνήμης E χρησιμοποιώντας τον τύπο:
E={for(i=0,i<numOfVPsThatUsedMemory)sum(r_i*t_i)}/(simulationTime*MemorySize)
όπου r_i και t_i είναι το μέγεθος της κάθε VP(r_i) και ο χρόνος που κατανάλωσε στη μνήμη(t_i) αντιστοιχα.
Επίσης υπολογίζω την μέση τιμή του μεγέθους των κενών μνήμης με τον τύπο:
M={ for (i=0, i<simulationTime sum(x_i) } / simulationTime
Όπου x_i είναι η κενή μνήμη το δευτερόλεπτο i.
Ακόμα υπολογίζω τη διακύμανση του μεγέθους των κενών μνήμης με τον τύπο:
V={ for (i=0, i<simulationTime sum((x_i - M)^2) } / simulationTime
Όπου x_i είναι η κενή μνήμη το δευτερόλεπτο i και M το αποτέλεσμα του παραπάνω τύπου.

Εκτέλεση και μεταγλώττιση:

Για την μεταγλώττιση του προγράμματος χρησιμοποιείται το makefile.
Για την εκτέλεση του προγράμματος θα πρεπει να εκτελεστεί εντολή της μορφής:
./generator lo hi t T S D MP
Όπου το τυχαίο μέγεθος(σε KB) της κάθε VP είναι ομοιόμορφα κατανεμημένο στο διάστημα [lo, hi].Οιι χρόνοι μεταξύ δύο διαδοχικών αφίξεων είναι ανεξάρτητοι μεταξύ τους και εκθετικά κατανεμημένοι με μέση διάρκεια τ.Ο χρόνος ζωής (sec) της κάθε VP είναι εκθετικά κατανεμημένος με μέση διάρκεια T.Η συνολική διάρκεια προσομοίωσης D(seconds) και το μέγεθος της μνήμης S(KB).Τα t και T ειναι τιμές πάνω από το 1.

Παρατηρήσεις από τις μετρήσεις:

Για memorySize=256KB και simulationTime=100seconds
Επομένως, σύμφωνα με τα παραπανω πειράματα βλέπουμε ότι κατα μέσο όρο το best-fit και το worst-fit εξυπηρετούν περισσότερες διεργασίες από το buddy ειδικά αν το άνω όριο μεγέθους διεργασιων hi ειναι κοντα στο μέγεθος της μνήμης. Αυτό γίνεται βέβαια επειδή αν βρεθει μια διεργασία με μέγεθος μεγαλυτερου του μισού της μνήμης συμφωνα με το buddy θα πρέπει να καταληφθεί όλη η μνήμη δηλαδή μπαίνει πιο δυσκολα και δε επιτρέπει σε αλλες μικρες διεργασίες να εκτελεστούν κατα την παραμονή της μεγάλης διεργασίας στη μνήμη.