

CEID

Λειτουργικά συστήματα

2o project 2024-2025

Αντωνίου Σωτήριος (1067512)

basotos@hotmail.com

Κωνσταντίνος Χριστάκος (1070903)

up1070903@ac.upatras.gr

1. Πρώτη Φάση: υποστήριξη πολλαπλών επεξεργαστών

1)FCFS:

Η FCFS με πολλούς επεξεργαστές μπορούσε να υλοποιηθεί με διάφορους τρόπους. Ένας απ'αυτούς ήταν η υλοποίηση με threads, εκ των οποίων το καθένα θα αντιπροσώπευε έναν επεξεργαστή οι οποίοι παράλληλα θα εκτελούσαν μία διαφορετική διεργασία ο καθένας. Θα παραθέσω ένα πρότυπο αρχείο για αυτήν την υλοποίηση που κάναμε στα αρχικά στάδια του project.

Παρ'όλα αυτά εν τέλει αποφασίσαμε να παραμείνουμε στο ίδιο μοτίβο, δημιουργώντας πολλά child functions, καθένα απ'τα οποία αντιπροσωπεύει έναν επεξεργαστή, και την δημιουργία pipes[] για την επικοινωνία του function FCFS και του καινούργιου function child_process (αυτό το function είναι το function που περιγράφει την λειτουργία κάθε επεξεργαστή. Στην αρχή της fcfs δημιουργούμε pipes[] και ένα child_process στιγμιότυπο για κάθε επεξεργαστή).

Οι επεξεργαστές αυτοί δεν έχουν διαφορετικό πατέρα ο καθένας (δεν έχουν γενικά πατέρα). Ενώ αυτά τρέχουν σε ένα while-loop κάνοντας τις διεργασίες που τους ανατίθενται, ο πραγματικός πατέρας πίσω από αυτό, που λαμβάνει σήματα απ'τον κάθε processor και στέλνει καινούργιες διεργασίες, είναι η συνέχεια της fcfs function.

Πιο λεπτομερώς αφού δημιουργούμε τους processors και τα ζευγάρια pipes (ένα για να στέλνει τις διεργασίες ο scheduler στους επεξεργαστές και ένα για να λαμβάνει feedback απ'αυτες), προχωράμε στο while loop του scheduler απ'το οποίο και δεν θα φύγει μέχρι να αδειάσει το queue με τις διεργασίες. Αν μια διεργασία χρειάζεται περισσότερα processors απ'όσα έχουμε ορίσει στο κάλεσμα, αυτή η διεργασία είναι αδύνατο να πραγματοποιηθεί σε οποιαδήποτε κατάσταση, οπότε την αναγνωρίζουμε στην main και δεν την βάζουμε καν στην ουρά.

Το πρώτο πράγμα που κάνουν οι "επεξεργαστές" είναι να μπουν σε ένα while-loop όπου στέλνουν το σήμα READY και περιμένουν να παραλάβουν μια διεργασία. Απο εδώ και πέρα οι processors λειτουργούν με παρόμοιο τρόπο όπως λειτουργούσαν στον αρχικό κώδικα. Παραλαμβάνουν μία διεργασία, την εκτελούν και εκτυπώνουν τους χρόνους με την μόνη διαφορά ότι όταν τελειώσουν θα στείλουν το σήμα DONE (θα δούμε στην FCFS γιατί αυτό είναι απαραίτητο) στο scheduler (FCFS FUNCTION) και θα περιμένουν να διαβαστεί για να αποφύγουμε race conditions, αφού αμέσως μετά από το σήμα DONE θα επαναλάβουν το

LOOP και θα στείλουν το σήμα READY.

Ο κύριος εγκέφαλος πίσω απ'το τί συμβαίνει είναι η while της fcfs. Πρώτα απ'όλα η FCFS δεν λειτουργεί reactively στα σήματα όπως οι processors, αλλά ελέγχει κάθε pipe με σειρά εξετάζοντας αν ήρθε κάποιο σήμα.

Αν το σήμα αυτο είναι DONE, θα αυξήσει, κατα οσους επεξεργαστές χρησιμοποιούνταν, την μεταβλητή available_processors στην οποία ανα πάσα στιγμή αποθηκεύει πόσους επεξεργαστές έχει διαθέσιμους

Αν το σήμα αυτό είναι READY, θα κάνει 1 την μεταβλητή που του δείχνει την κατάσταση καθε processor: processors_ready[i].

Αν έχει πάνω απο ένα διαθέσιμο επεξεργαστή, αυτό σημαίνει ότι θα προσπαθήσει να του αναθέσει μια δουλειά.

Η αναθεση γίνεται με παρόμοιο τρόπο με το αρχικό πρόγραμμα, αλλά αλλάζουμε τις μεταβλητές processors_ready[i] available_processors αφού στείλαμε δουλειά στον processor.

Αν η λίστα δεν έχει άλλες διεργασίες, θα κλείσει τα pipes και οι επεξεργαστές θα απενεργοποιηθούν (θα βγουν απ'το loop).

Αν έχει, θα την βάλει στο proc. θα ελέγξει την μεταβλητή processors_required του proc (καινούρια μεταβλητή που δείχνει την ανάγκη καθε διεργασίες για επεξεργαστές).

Αν είναι αρκετοί διαθέσιμοι θα γίνει ανάθεση, μειώνοντας τον αριθμό των διαθέσιμων επεξεργαστών κατα τόσους όσους χρειάζεται. Η ανάθεση πρακτικά γίνεται σε έναν επεξεργαστή, αλλά από την κατανόηση μου πάνω στην άσκηση, καταλαβαίνω οτι και ο άλλος επεξεργαστής πρέπει να μείνει inactive αυτό το διάστημα.

Αν δεν είναι αρκετοί διαθέσιμοι, θα βάλουμε την διεργασία στο τέλος της ουράς και θα επαναλάβουμε τον κύκλο για να βρούμε την επόμενη διεργασία

παραθέτω ένα παράδειγμα οπου τρέχω το παρακάτω reverse.txt με 3 επεξεργαστές:

```
../work/work7 2
../work/work6 2
../work/work5
../work/work4 1
../work/work3 3
../work/work2 4
../work/work1
```

```

➔ scheduler ./scheduler_v2 FCFS 3 reverse.txt
Error: Process ../work/work2 requires 4 processors, but only 3 are available. Skipping.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Processor 0 is ready.
[DEBUG] Assigning process ../work/work7 to processor 0. Available processors: 1
[DEBUG] Processor 1 is ready.
[DEBUG] Assigning process ../work/work5 to processor 1. Available processors: 0
[DEBUG] Processor 0 received process ../work/work7 (requires 2 processors).
[DEBUG] Processor 1 received process ../work/work5 (requires 1 processors).
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Processor 2 is ready.
[DEBUG] Processor 0 starting execution of ../work/work7.
[DEBUG] Processor 1 starting execution of ../work/work5.
process 24163 begins
process 24164 begins
process 24164 ends
Processor 1 completed process ../work/work5.
Execution time = 30.14 secs
Elapsed time = 30.14 secs
Workload time = 30.14 secs
[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 1.
[DEBUG] Assigning process ../work/work4 to processor 2. Available processors: 0
[DEBUG] Processor 1 received acknowledgment from parent.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Processor 1 is ready.
[DEBUG] Processor 2 received process ../work/work4 (requires 1 processors).
[DEBUG] Processor 2 starting execution of ../work/work4.
process 24175 begins
process 24163 ends
Processor 0 completed process ../work/work7.
Execution time = 41.60 secs
Elapsed time = 41.61 secs
Workload time = 41.61 secs
[DEBUG] Processor 0 sent DONE signal.
[DEBUG] Processor 0 completed a task. Available processors: 2
[DEBUG] Parent sent acknowledgment to processor 0.
[DEBUG] Assigning process ../work/work1 to processor 1. Available processors: 1
[DEBUG] Processor 0 received acknowledgment from parent.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Processor 1 received process ../work/work1 (requires 1 processors).
[DEBUG] Processor 0 is ready.
[DEBUG] Processor 1 starting execution of ../work/work1.
process 24182 begins
process 24182 ends
Processor 1 completed process ../work/work1.
Execution time = 6.24 secs

```

```

process 24182 begins
process 24182 ends
Processor 1 completed process ../work/work1.
Execution time = 6.24 secs
Elapsed time = 47.85 secs
Workload time = 47.85 secs
[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 2
[DEBUG] Processor 1 received acknowledgment from parent.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Parent sent acknowledgment to processor 1.
[DEBUG] Assigning process ../work/work6 to processor 0. Available processors: 0
[DEBUG] Processor 0 received process ../work/work6 (requires 2 processors).
[DEBUG] Processor 1 is ready.
[DEBUG] Processor 0 starting execution of ../work/work6.
process 24187 begins
process 24175 ends
Processor 2 completed process ../work/work4.
Execution time = 25.52 secs
Elapsed time = 55.66 secs
Workload time = 55.66 secs
[DEBUG] Processor 2 sent DONE signal.
[DEBUG] Processor 2 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 2.
[DEBUG] Processor 2 received acknowledgment from parent.
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Processor 2 is ready.
process 24187 ends
Processor 0 completed process ../work/work6.
Execution time = 29.65 secs
Elapsed time = 77.50 secs
Workload time = 77.50 secs
[DEBUG] Processor 0 sent DONE signal.
[DEBUG] Processor 0 completed a task. Available processors: 3
[DEBUG] Parent sent acknowledgment to processor 0.
[DEBUG] Assigning process ../work/work3 to processor 1. Available processors: 0
[DEBUG] Processor 0 received acknowledgment from parent.
[DEBUG] Processor 1 received process ../work/work3 (requires 3 processors).
[DEBUG] Processor 0 is ready.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Processor 1 starting execution of ../work/work3.
process 24200 begins
process 24200 ends
Processor 1 completed process ../work/work3.
Execution time = 14.32 secs
Elapsed time = 91.82 secs
Workload time = 91.82 secs
[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 3

```

```

[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 3
[DEBUG] Processor 1 received acknowledgment from parent.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Parent sent acknowledgment to processor 1.
[DEBUG] Processor 2 received no more processes. Exiting.
[DEBUG] Processor 1 is ready.
[DEBUG] Processor 1 received no more processes. Exiting.
[DEBUG] Processor 0 received no more processes. Exiting.
All processors finished execution.
WORKLOAD TIME: 91.82 secs
Scheduler exits.
→ scheduler █

```

προβλήματα:

- 1) προβλήματα συγχρονισμού μεταξύ των ready-done σημάτων μας ανάγκασε να περιμένουμε απάντηση απ'τον γονέα για να μην αντιμετωπίζουμε race-conditions.
- 2) Πολλα προβλήματα κατανόησης στο πως λειτουργεί το πρόγραμμα, ποια θα είναι η ιδανική υλοποίηση των ζητημάτων, και στον τρόπο υλοποίησης. Μικρά προβλήματα για να μπορέσω να απαριθμήσω, αλλά στην γενική εικόνα μας έφαγαν τον μεγαλύτερο χρόνο της άσκησης.
- 3) Η μεταβλητή `available_processors` ήταν εκ των υστέρων προσθήκη για βασικά προβλήματα που δημιουργόντουσαν, όπως ότι δεν μπορούσαμε να ξέρουμε πόσοι επεξεργαστές δουλεύουν για την ανάθεση μιας δουλειάς που απαιτεί πολλαπλούς επεξεργαστές.
- 4) Η μεταβλητή `processor_ready[i]` ήταν επίσης εκ των υστέρων προσθήκη αφού ένα μήνυμα σε `pipe` δεν παραμένει αφού διαβαστεί οπότε έπρεπε να αποθηκεύουμε την κατάσταση του κάθε επεξεργαστή.
- 5) Υπήρχε ένα μπέρδεμα με την εκφώνηση της άσκησης που κάθε διεργασία εκτελείται μόνο από έναν επεξεργαστή, αλλά μπορεί να ζητάει πολλούς.

2) ROUND ROBIN

Η υλοποίηση που κάναμε για τον round robin ήταν παρόμοια. Για την εξομοίωση πολλών επεξεργαστών χρησιμοποιήσαμε `for loops`. Κάθε επεξεργαστής έχει το δικό του `i`. Η διαφοράς με τον αρχικό κώδικα είναι λίγες. Μία βασική είναι ότι πλέον το `proc_running_process` είναι πλέον `table (running_processes[i])` καθώς πρέπει να μπορούμε να διατηρούμε πολλές διεργασίες που γίνονται ταυτόχρονα σε μια μεταβλητή και να πειράζουμε τα δεδομένα τους. Η αλλαγή έγινε και στο `function sigchld_handler` για όμοιους σκοπούς. Για να ξέρουμε αν έχουμε αρκετούς επεξεργαστές για μια διεργασία, όπως στην 1 χρησιμοποιήσαμε `available_processors` μεταβλητή η οποία αυξάνεται κατά 1 κάθε φορά που ένα `process` σταματάει (είτε λόγω χρόνου είτε επειδή τελείωσε) και μειώνεται κατά 1 κάθε φορά που ένας επεξεργαστής αναλαμβάνει μία διεργασία. Όταν μία διεργασία δεν μπορεί να διεξαχθεί λόγω περιορισμένων πόρων, ο επεξεργαστής ψάχνει την επόμενη διεργασία που μπορεί να εκτελέσει.

Η λογική είναι η ίδια με τον πρώτη άσκηση, με την μεγαλύτερη διαφορά να είναι ότι πλέον έχουμε ένα λούπ που διαβάζει 1-1 τα στάτους των διεργασιών και ανταποκρίνεται, αλλάζοντας το στάτους, ή δίνοντας τους δουλειά και έτσι δημιουργούνται πολλά `fork` αντιπροσωπεύοντας πολλούς επεξεργαστές.

Η τελευταία προσθήκη που έπρεπε να γίνει είναι ότι η `sigchild_handler` λαμβάνει μόνο το `pid` της εργασίας και πλέον δεν έχει τα στοιχεία του ποια λειτουργία τερμάτισε αφού οι ταυτόχρονες λειτουργίες είναι πολλές. Για αυτο επιβαλλόταν να προσθέσουμε ένα λουπ ώστε να ταιριάζουμε το `pid` της εκάστοτε εργασίας που τερμάτισε με τον επεξεργαστή στον οποίο εκτελούνταν, και κατ'επέκταση το `i` της `running_processes[i]`.

παράδειγμα υλοποίησης με 4 επεξεργαστές για κβάντο χρόνου 5000:

```
../work/work7 1
../work/work6 7
../work/work5 3
../work/work4 2
../work/work3
../work/work2 3
../work/work1
```

```
➔ scheduler ./scheduler_v4 RR 5000 reverse.txt
Error: Process ../work/work6 requires 7 processors, but only 4 are available. Skipping.
Processor 0 executing ../work/work7
Processor 1 executing ../work/work5
Processor 2 executing ../work/work4
Processor 3 executing ../work/work3
process 40576 begins
process 40577 begins
process 40578 begins
process 40579 begins
Processor 0 stopping ../work/work7 (PID: 40576)
Processor 1 stopping ../work/work5 (PID: 40577)
Processor 2 stopping ../work/work4 (PID: 40578)
Processor 3 stopping ../work/work3 (PID: 40579)
Processor 2 resuming ../work/work7 (PID: 40576)
Processor 3 resuming ../work/work4 (PID: 40578)
Processor 0 executing ../work/work2
Processor 1 executing ../work/work1
process 40583 begins
process 40584 begins
Processor 0 stopping ../work/work2 (PID: 40583)
Processor 1 stopping ../work/work1 (PID: 40584)
Processor 2 stopping ../work/work7 (PID: 40576)
Processor 3 stopping ../work/work4 (PID: 40578)
Processor 0 resuming ../work/work3 (PID: 40579)
Processor 1 resuming ../work/work5 (PID: 40577)
Processor 2 resuming ../work/work2 (PID: 40583)
Processor 3 resuming ../work/work1 (PID: 40584)
process 40584 ends
SIGCHLD received for PID 40584
processor 3 has -1 processes
Processor 3 - PID 40584 - CMD: ../work/work1
    Elapsed time = 13.70 secs
    Execution time = 8.67 secs
    Workload time = 13.70 secs
Processor 0 stopping ../work/work3 (PID: 40579)
Processor 1 stopping ../work/work5 (PID: 40577)
Processor 2 stopping ../work/work2 (PID: 40583)
Processor 0 resuming ../work/work7 (PID: 40576)
Processor 1 resuming ../work/work4 (PID: 40578)
Processor 2 resuming ../work/work3 (PID: 40579)
Processor 3 resuming ../work/work5 (PID: 40577)
Processor 0 stopping ../work/work7 (PID: 40576)
Processor 1 stopping ../work/work4 (PID: 40578)
Processor 2 stopping ../work/work3 (PID: 40579)
Processor 3 stopping ../work/work5 (PID: 40577)
Processor 0 resuming ../work/work2 (PID: 40583)
Processor 1 resuming ../work/work7 (PID: 40576)
Processor 2 resuming ../work/work4 (PID: 40578)
```

```

Processor 1 stopping ../work/work4 (PID: 40578)
Processor 2 stopping ../work/work3 (PID: 40579)
Processor 3 stopping ../work/work5 (PID: 40577)
Processor 0 resuming ../work/work2 (PID: 40583)
Processor 1 resuming ../work/work7 (PID: 40576)
Processor 2 resuming ../work/work4 (PID: 40578)
Processor 3 resuming ../work/work3 (PID: 40579)
Processor 0 stopping ../work/work2 (PID: 40583)
Processor 1 stopping ../work/work7 (PID: 40576)
Processor 2 stopping ../work/work4 (PID: 40578)
Processor 3 stopping ../work/work3 (PID: 40579)
Processor 0 resuming ../work/work5 (PID: 40577)
Processor 1 resuming ../work/work2 (PID: 40583)
Processor 2 resuming ../work/work7 (PID: 40576)
Processor 3 resuming ../work/work4 (PID: 40578)
Processor 0 stopping ../work/work5 (PID: 40577)
Processor 1 stopping ../work/work2 (PID: 40583)
Processor 2 stopping ../work/work7 (PID: 40576)
Processor 3 stopping ../work/work4 (PID: 40578)
Processor 0 resuming ../work/work3 (PID: 40579)
Processor 1 resuming ../work/work5 (PID: 40577)
Processor 2 resuming ../work/work2 (PID: 40583)
Processor 3 resuming ../work/work7 (PID: 40576)
process 40583 ends
SIGCHLD received for PID 40583
processor 2 has -1 processes
Processor 2 - PID 40583 - CMD: ../work/work2
Elapsed time = 31.66 secs
Execution time = 26.63 secs
Workload time = 31.66 secs
Processor 0 stopping ../work/work3 (PID: 40579)
Processor 1 stopping ../work/work5 (PID: 40577)
Processor 3 stopping ../work/work7 (PID: 40576)
Processor 0 resuming ../work/work4 (PID: 40578)
Processor 1 resuming ../work/work3 (PID: 40579)
Processor 2 resuming ../work/work5 (PID: 40577)
Processor 3 resuming ../work/work7 (PID: 40576)
Processor 0 stopping ../work/work4 (PID: 40578)
Processor 1 stopping ../work/work3 (PID: 40579)
Processor 2 stopping ../work/work5 (PID: 40577)
Processor 3 stopping ../work/work7 (PID: 40576)
Processor 0 resuming ../work/work4 (PID: 40578)
Processor 1 resuming ../work/work3 (PID: 40579)
Processor 2 resuming ../work/work5 (PID: 40577)
Processor 3 resuming ../work/work7 (PID: 40576)
process 40579 ends
SIGCHLD received for PID 40579
processor 1 has -1 processes
Processor 1 - PID 40579 - CMD: ../work/work3
SIGCHLD received for PID 40579
processor 1 has -1 processes
Processor 1 - PID 40579 - CMD: ../work/work3
Elapsed time = 37.02 secs
Execution time = 37.00 secs
Workload time = 37.02 secs
Processor 0 stopping ../work/work4 (PID: 40578)
Processor 2 stopping ../work/work5 (PID: 40577)
Processor 3 stopping ../work/work7 (PID: 40576)
Processor 0 resuming ../work/work4 (PID: 40578)
Processor 1 resuming ../work/work5 (PID: 40577)
Processor 2 resuming ../work/work7 (PID: 40576)
process 40578 ends
SIGCHLD received for PID 40578
processor 0 has -1 processes
Processor 0 - PID 40578 - CMD: ../work/work4
Elapsed time = 41.70 secs
Execution time = 41.68 secs
Workload time = 41.70 secs
Processor 1 stopping ../work/work5 (PID: 40577)
Processor 2 stopping ../work/work7 (PID: 40576)
Processor 0 resuming ../work/work5 (PID: 40577)
Processor 1 resuming ../work/work7 (PID: 40576)
Processor 0 stopping ../work/work5 (PID: 40577)
Processor 1 stopping ../work/work7 (PID: 40576)
Processor 0 resuming ../work/work5 (PID: 40577)
Processor 1 resuming ../work/work7 (PID: 40576)
process 40577 ends
SIGCHLD received for PID 40577
processor 0 has -2 processes
Processor 0 - PID 40577 - CMD: ../work/work5
Elapsed time = 49.18 secs
Execution time = 49.17 secs
Workload time = 49.18 secs
Processor 1 stopping ../work/work7 (PID: 40576)
Processor 0 resuming ../work/work7 (PID: 40576)
process 40576 ends
SIGCHLD received for PID 40576
processor 0 has -3 processes
Processor 0 - PID 40576 - CMD: ../work/work7
Elapsed time = 53.92 secs
Execution time = 53.92 secs
Workload time = 53.92 secs
WORKLOAD TIME: 53.92 secs
Scheduler exits.
→ scheduler

```

προβλήματα: Βοηθηθήκαμε αρκετά απ'την υλοποίηση του FCFS και έχοντας πλέον κατανοήσει πως λειτουργεί το πρόγραμμα και τι θέλουμε να υλοποιήσουμε (π.χ. η λειτουργία των `available_processors` είναι ίδια με το πρώτο μέρος), δεν αντιμετωπίσαμε ιδιαίτερες δυσκολίες. Το πιο δύσκολο κομμάτι ήταν η ιδέα ότι μπορούμε να χρησιμοποιήσουμε ένα `table` απ'το `struct` που έχουμε στο οποίο θα αποθηκεύονται όλες οι πληροφορίες των ταυτόχρονα εκτελούμενων διεργασιών.

2) Έπρεπε να αφήσουμε την τεχνική που ο χρήστης έδινε τον αριθμό των επεξεργασιών στον `scheduler` αφού τώρα το χρειαζόμαστε ως `global value`, και απ'την εκφώνηση δεν μπορέσαμε να καταλάβουμε ποια είναι η ζητούμενη υλοποίηση απ'τις 2.

3) RR-AFF

Για το τελευταίο κομμάτι του `project` έπρεπε αρχικά να δημιουργήσουμε μια καινούργια μεταβλητή στο `proc_desc` μας που θα αποθηκεύει τον επεξεργαστή που

δούλεψε αρχικά σαυτό το task. Όταν κάθε επεξεργαστής αναλαμβάνει μια διεργασία για πρώτη φορά, βάζει το `i/id` του στην μεταβλητή αυτή. Στην συνέχεια κάθε φορά που ένας επεξεργαστής προσπαθεί να συνεχίσει (SIGCONT) την διεργασία αυτή συγκρίνει το `id` του με το `id` που έχει αυτή η διεργασία, και αν είναι διαφορετικό θα την βάλει στο τέλος του queue, και θα προσπαθήσει να πάρει την επόμενη.

Για να ξέρει ένας επεξεργαστής ότι τελείωσε και δεν έχει άλλες διεργασίες να πάρει, κρατάει σε μία μεταβλητή το πόσες διεργασίες έχει πάρει, την οποία αυξάνει κάθε φορά που παίρνει μία καινούργια και μειώνει κάθε φορά που μια εργασία τερματίζει (η SIGCHLD_HANDLER είναι υπεύθυνη για την μείωση της μεταβλητής αυτής)

Η υπόλοιπη υλοποίηση είναι ίδια με την RR. Ως δεύτερη λύση σκεφτήκαμε ότι οι ίδιες οι διεργασίες θα μπορούσαν να συγκρίνονται με διαφορετικούς επεξεργαστές για το ποιες διεργασίες μπορεί να παραλάβει ο καθένας, αλλά αυτό θα μπορούσε να μπλοκάρει το πρόγραμμα, δημιουργώντας καταστάσεις στις οποίες δεν χρησιμοποιούμε όλους τους επεξεργαστές για το συγκεκριμένο κβάντο χρόνου, αλλά λίγότερους, αν μία διεργασία χρειαζόταν επεξεργαστή ο οποίος ήδη υλοποιούσε άλλη διεργασία.

παράδειγμα εκτέλεσης με 4 επεξεργαστές και 5000 κβάντο:

```
../work/work7 1
../work/work6 7
../work/work5 3
../work/work4 2
../work/work3
../work/work2 3
../work/work1
```

```
Error: Process ../work/work6 requires 7 processors, but only 4 are available. Skipping.
processor 0 has 1 processes
processor 1 has 1 processes
processor 2 has 1 processes
Processor 0 executing ../work/work7
processor 3 has 1 processes
Processor 1 executing ../work/work5
Processor 2 executing ../work/work4
process 41118 begins
Processor 3 executing ../work/work3
process 41120 begins
process 41119 begins
process 41121 begins
Processor 0 stopping ../work/work7 (PID: 41118)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 3 stopping ../work/work3 (PID: 41121)
processor 0 has 2 processes
processor 1 has 2 processes
Processor 2 resuming ../work/work4 (PID: 41120)
Processor 3 resuming ../work/work3 (PID: 41121)
Processor 0 executing ../work/work2
Processor 1 executing ../work/work1
process 41126 begins
process 41127 begins
Processor 0 stopping ../work/work2 (PID: 41126)
Processor 1 stopping ../work/work1 (PID: 41127)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 3 stopping ../work/work3 (PID: 41121)
Processor 0 resuming ../work/work7 (PID: 41118)
Processor 1 resuming ../work/work5 (PID: 41119)
Processor 2 resuming ../work/work4 (PID: 41120)
Processor 3 resuming ../work/work3 (PID: 41121)
Processor 0 stopping ../work/work7 (PID: 41118)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 3 stopping ../work/work3 (PID: 41121)
Processor 0 resuming ../work/work2 (PID: 41126)
Processor 1 resuming ../work/work1 (PID: 41127)
Processor 2 resuming ../work/work4 (PID: 41120)
Processor 3 resuming ../work/work3 (PID: 41121)
process 41127 ends
SIGCHLD received for PID 41127
processor 1 has 1 processes
Processor 1 - PID 41127 - CMD: ../work/work1
Elapsed time = 18.14 secs
Execution time = 13.13 secs
Workload time = 18.14 secs
```

```

processor 1 has 1 processes
Processor 1 - PID 41127 - CMD: ../work/work1
Elapsed time = 18.14 secs
Execution time = 13.13 secs
Workload time = 18.14 secs
Processor 0 stopping ../work/work2 (PID: 41126)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 3 stopping ../work/work3 (PID: 41121)
Processor 0 resuming ../work/work7 (PID: 41118)
Processor 1 resuming ../work/work5 (PID: 41119)
Processor 2 resuming ../work/work4 (PID: 41120)
Processor 3 resuming ../work/work3 (PID: 41121)
Processor 0 stopping ../work/work7 (PID: 41118)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 3 stopping ../work/work3 (PID: 41121)
Processor 0 resuming ../work/work2 (PID: 41126)
Processor 1 resuming ../work/work5 (PID: 41119)
Processor 2 resuming ../work/work4 (PID: 41120)
Processor 3 resuming ../work/work3 (PID: 41121)
process 41121 ends
SIGCHLD received for PID 41121
processor 3 has 0 processes
Processor 3 - PID 41121 - CMD: ../work/work3
Elapsed time = 24.25 secs
Execution time = 24.24 secs
Workload time = 24.25 secs
Processor 0 stopping ../work/work2 (PID: 41126)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 0 resuming ../work/work7 (PID: 41118)
Processor 1 resuming ../work/work5 (PID: 41119)
Processor 2 resuming ../work/work4 (PID: 41120)
Processor 0 stopping ../work/work7 (PID: 41118)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 2 stopping ../work/work4 (PID: 41120)
Processor 0 resuming ../work/work2 (PID: 41126)
Processor 1 resuming ../work/work5 (PID: 41119)
Processor 2 resuming ../work/work4 (PID: 41120)
process 41120 ends
SIGCHLD received for PID 41120
processor 2 has 0 processes
Processor 2 - PID 41120 - CMD: ../work/work4
Elapsed time = 30.53 secs
Execution time = 30.53 secs
Workload time = 30.53 secs
Processor 0 stopping ../work/work2 (PID: 41126)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 0 resuming ../work/work7 (PID: 41118)

processor 2 has 0 processes
Processor 2 - PID 41120 - CMD: ../work/work4
Elapsed time = 30.53 secs
Execution time = 30.53 secs
Workload time = 30.53 secs
Processor 0 stopping ../work/work2 (PID: 41126)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 0 resuming ../work/work7 (PID: 41118)
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 0 resuming ../work/work2 (PID: 41126)
Processor 1 resuming ../work/work5 (PID: 41119)
process 41126 ends
SIGCHLD received for PID 41126
processor 0 has 1 processes
Processor 0 - PID 41126 - CMD: ../work/work2
Elapsed time = 38.83 secs
Execution time = 33.82 secs
Workload time = 38.83 secs
Processor 1 stopping ../work/work5 (PID: 41119)
Processor 0 resuming ../work/work7 (PID: 41118)
Processor 1 resuming ../work/work5 (PID: 41119)
process 41119 ends
SIGCHLD received for PID 41119
processor 1 has 0 processes
Processor 1 - PID 41119 - CMD: ../work/work5
Elapsed time = 40.46 secs
Execution time = 40.46 secs
Workload time = 40.46 secs
Processor 0 stopping ../work/work7 (PID: 41118)
Processor 0 resuming ../work/work7 (PID: 41118)
Processor 0 stopping ../work/work7 (PID: 41118)
Processor 0 resuming ../work/work7 (PID: 41118)
process 41118 ends
SIGCHLD received for PID 41118
processor 0 has 0 processes
Processor 0 - PID 41118 - CMD: ../work/work7
Elapsed time = 50.13 secs
Execution time = 50.13 secs
Workload time = 50.13 secs
WORKLOAD TIME: 50.13 secs
Scheduler exits.
→ scheduler

```

προβλήματα:

το μεγαλύτερο πρόβλημα που αντιμετωπίσαμε ήταν το πόσο συχνά πέφταμε σε endless loop, αφού ένας επεξεργαστής μπορεί να έψαχνε συνεχώς για μια διεργασία που μπορούσε να υλοποιήσει, χωρίς όμως να γνωρίζει ότι είχε υλοποιήσει όλες του τις εργασίες. Το πρόβλημα αυτό λύθηκε όταν αρχίσαμε να κάνουμε track τις διεργασίες που έχει ήδη υλοποιήσει και αυτές που περιμένουν να υλοποιηθούν. Πέρα απ' αυτό η υπόλοιπη άσκηση έμοιαζε πάρα πολύ με την 1.2: RR

2. Δεύτερη Φάση: υποστήριξη αιτήσεων πολλαπλών επεξεργαστών

Η μόνη διαφορά απ' το πρώτο βήμα είναι στον διαμοιρασμό των διεργασιών. Τα θεμέλια της υλοποίησης έχουν ήδη φτιαχτεί απο το 1ο ερώτημα, και το μόνο που έμενε να καταφέρουμε

είναι να μπορέσουμε να μοιράσουμε την ίδια διεργασία σε πολλούς επεξεργαστές. Ο τρόπος να το καταφέρουμε αυτό βασίζεται στο τι κάνει η κάθε διεργασία.

Συγκεκριμένα είδαμε ότι ο χρόνος κάθε διεργασίας βασίζεται πάρα πολύ στην πράξη $WORKLOAD * DELAY$ όπου το $WORKLOAD$ είναι ο αριθμός στο τέλος κάθε executable. Αυτό που έμενε είναι να βάλουμε έναν έλεγχο όπου πλέον όταν η FCFS μας πει να στείλει μια διεργασία που απαιτεί πολλούς επεξεργαστές, αντί να την στείλει την μεταφέρει στο καινούργιο μας FUNCTION: `split_task` (αν θέλει 1 επεξεργαστή, τα βήματα που ακολουθούμε είναι τα ίδια με την 1η φάση)

Εκεί πέρα διαβάζουμε το $WORKLOAD$ και το μοιράζουμε σε κομμάτια = τον αριθμο επεξεργαστών που θα εκτελέσουν την διεργασία. ΠΧ η διεργασία `work7` σε 3 επεξεργαστες θα γίνει `work2 work2 και work3`.

Εν συνέχεια οι καινούργιες διεργασίες θα μπουν στην αρχή της σειράς ως μικρότερες διεργασίες / μέλη μίας μεγαλύτερης διεργασίας, όπου πλέον υλοποιούνται από έναν μόνο επεξεργαστή και η διαδικασία υλοποίησης παραμένει η ίδια.

```
../work/work7 2
../work/work6
../work/work5
../work/work4 1
../work/work3 3
../work/work2 4
../work/work1
```

```

➤ scheduler ./scheduler_v3 FCFS 3 reverse.txt
Error: Process ../work/work2 requires 4 processors, but only 3 are available. Skipping.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Processor 0 is ready.
[DEBUG] Splitting task: ../work/work7 (PID: -1, Processors: 2)
[DEBUG] Created subtask: ../work/work4 (Sub-PID: 0, Workload: 4)
[DEBUG] Created subtask: ../work/work3 (Sub-PID: 1, Workload: 3)
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Processor 1 is ready.
[DEBUG] Assigning process ../work/work3 to processor 1 with pid: 1. Available processors: 2
[DEBUG] Processor 2 is ready.
[DEBUG] Assigning process ../work/work4 to processor 2 with pid: 0. Available processors: 1
[DEBUG] Processor 1 received process ../work/work3 (requires 1 processors).
[DEBUG] Assigning process ../work/work6 to processor 0 with pid: -1. Available processors: 0
[DEBUG] Processor 2 received process ../work/work4 (requires 1 processors).
[DEBUG] Processor 0 received process ../work/work6 (requires 1 processors).
[DEBUG] Processor 0 starting execution of ../work/work6.
[DEBUG] Processor 1 starting execution of ../work/work3.
[DEBUG] Processor 2 starting execution of ../work/work4.
process 28482 begins
process 28481 begins
process 28483 begins
process 28481 ends
Processor 1 completed process ../work/work3 with pid 28481.
Execution time = 21.73 secs
Elapsed time = 21.73 secs
Workload time = 21.73 secs
[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 1.
[DEBUG] Processor 1 received acknowledgment from parent.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Processor 1 is ready.
[DEBUG] Assigning process ../work/work5 to processor 1 with pid: -1. Available processors: 0
[DEBUG] Processor 1 received process ../work/work5 (requires 1 processors).
[DEBUG] Processor 1 starting execution of ../work/work5.
process 28491 begins
process 28483 ends
Processor 2 completed process ../work/work4 with pid 28483.
Execution time = 26.99 secs
Elapsed time = 26.99 secs
Workload time = 26.99 secs
[DEBUG] Processor 2 sent DONE signal.
[DEBUG] Processor 2 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 2.
[DEBUG] Processor 2 received acknowledgment from parent.
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Processor 2 is ready.

[DEBUG] Processor 2 sent DONE signal.
[DEBUG] Processor 2 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 2.
[DEBUG] Processor 2 received acknowledgment from parent.
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Processor 2 is ready.
[DEBUG] Assigning process ../work/work4 to processor 2 with pid: -1. Available processors: 0
[DEBUG] Processor 2 received process ../work/work4 (requires 1 processors).
[DEBUG] Processor 2 starting execution of ../work/work4.
process 28495 begins
process 28482 ends
Processor 0 completed process ../work/work6 with pid 28482.
Execution time = 40.41 secs
Elapsed time = 40.41 secs
Workload time = 40.41 secs
[DEBUG] Processor 0 sent DONE signal.
[DEBUG] Processor 0 completed a task. Available processors: 1
[DEBUG] Processor 0 received acknowledgment from parent.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Parent sent acknowledgment to processor 0.
[DEBUG] Processor 0 is ready.
[DEBUG] Assigning process ../work/work1 to processor 0 with pid: -1. Available processors: 0
[DEBUG] Processor 0 received process ../work/work1 (requires 1 processors).
[DEBUG] Processor 0 starting execution of ../work/work1.
process 28505 begins
process 28505 ends
Processor 0 completed process ../work/work1 with pid 28505.
Execution time = 6.09 secs
Elapsed time = 46.51 secs
Workload time = 46.51 secs
[DEBUG] Processor 0 sent DONE signal.
[DEBUG] Processor 0 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 0.
[DEBUG] Processor 0 received acknowledgment from parent.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Processor 0 is ready.
process 28495 ends
Processor 2 completed process ../work/work4 with pid 28495.
Execution time = 27.04 secs
Elapsed time = 54.03 secs
Workload time = 54.03 secs
[DEBUG] Processor 2 sent DONE signal.
[DEBUG] Processor 2 completed a task. Available processors: 2
[DEBUG] Parent sent acknowledgment to processor 2.
[DEBUG] Processor 2 received acknowledgment from parent.
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Processor 2 is ready.
process 28491 ends
Processor 1 completed process ../work/work5 with pid 28491.

```

```

process 28491 ends
  Processor 1 completed process ../work/work5 with pid 28491.
  Execution time = 32.76 secs
  Elapsed time = 54.49 secs
  Workload time = 54.49 secs
[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 3
[DEBUG] Parent sent acknowledgment to processor 1.
[DEBUG] Splitting task: ../work/work3 (PID: -1, Processors: 3)
[DEBUG] Processor 1 received acknowledgment from parent.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Created subtask: ../work/work1 (Sub-PID: 0, Workload: 1)
[DEBUG] Created subtask: ../work/work1 (Sub-PID: 1, Workload: 1)
[DEBUG] Created subtask: ../work/work1 (Sub-PID: 2, Workload: 1)
[DEBUG] Assigning process ../work/work1 to processor 0 with pid: 2. Available processors: 2
[DEBUG] Processor 1 is ready.
[DEBUG] Assigning process ../work/work1 to processor 1 with pid: 1. Available processors: 1
[DEBUG] Assigning process ../work/work1 to processor 2 with pid: 0. Available processors: 0
[DEBUG] Processor 0 received process ../work/work1 (requires 1 processors).
[DEBUG] Processor 1 received process ../work/work1 (requires 1 processors).
[DEBUG] Processor 2 received process ../work/work1 (requires 1 processors).
[DEBUG] Processor 1 starting execution of ../work/work1.
[DEBUG] Processor 0 starting execution of ../work/work1.
[DEBUG] Processor 2 starting execution of ../work/work1.
process 28517 begins
process 28516 begins
process 28515 begins
process 28515 ends
  Processor 1 completed process ../work/work1 with pid 28515.
  Execution time = 13.38 secs
  Elapsed time = 13.38 secs
  Workload time = 67.87 secs
[DEBUG] Processor 1 sent DONE signal.
[DEBUG] Processor 1 completed a task. Available processors: 1
[DEBUG] Parent sent acknowledgment to processor 1.
[DEBUG] Processor 1 received acknowledgment from parent.
[DEBUG] Processor 1 sent READY signal.
[DEBUG] Processor 1 is ready.
process 28516 ends
  Processor 0 completed process ../work/work1 with pid 28516.
  Execution time = 13.40 secs
  Elapsed time = 13.40 secs
  Workload time = 67.89 secs
[DEBUG] Processor 0 sent DONE signal.
[DEBUG] Processor 0 completed a task. Available processors: 2
[DEBUG] Processor 0 received acknowledgment from parent.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Parent sent acknowledgment to processor 0.
[DEBUG] Processor 0 is ready.

```

```

[DEBUG] Processor 0 sent DONE signal.
[DEBUG] Processor 0 completed a task. Available processors: 2
[DEBUG] Processor 0 received acknowledgment from parent.
[DEBUG] Processor 0 sent READY signal.
[DEBUG] Parent sent acknowledgment to processor 0.
[DEBUG] Processor 0 is ready.
process 28517 ends
  Processor 2 completed process ../work/work1 with pid 28517.
  Execution time = 13.57 secs
  Elapsed time = 13.57 secs
  Workload time = 68.06 secs
[DEBUG] Processor 2 sent DONE signal.
[DEBUG] Processor 2 completed a task. Available processors: 3
[DEBUG] Processor 2 received acknowledgment from parent.
[DEBUG] Processor 2 sent READY signal.
[DEBUG] Parent sent acknowledgment to processor 2.
[DEBUG] Processor 2 is ready.
[DEBUG] Processor 2 received no more processes. Exiting.
[DEBUG] Processor 1 received no more processes. Exiting.
[DEBUG] Processor 0 received no more processes. Exiting.
All processors finished execution.
WORKLOAD TIME: 68.06 secs
Scheduler exits.
→ scheduler

```

προβλήματα:

Πέρα απ'την κατανόηση του πως δημιουργούνται οι διεργασίες και πως θα τις μοιράσουμε, η υλοποίηση μετά απ'αυτό ήταν απλή.