
ΑΝΑΦΟΡΑ ΕΡΓΑΣΤΗΡΙΑΚΗΣ: “PART II”

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ
(CEID1135)

Παραδοχές

- Ως χρόνο εκτέλεσης των διεργασιών χρησιμοποιούμε την κατάληξη του εκτελέσιμου αρχείου (π.χ. αν το αρχείο λέγεται work3 τότε ο χρόνος εκτέλεσης θεωρούμε ότι θα είναι 3 δευτερόλεπτα).
- Στο αρχείο εισόδου ο αριθμός που ακολουθεί το μονοπάτι του εκτελέσιμου αρχείου work θεωρούμε ότι είναι η προτεραιότητα και τον χρησιμοποιούμε μόνο στην συνάρτηση PRIO (π.χ. αν μια γραμμή του αρχείου εισόδου είναι η “../work/work75” τότε το 5 είναι η προτεραιότητα και το 7 ο χρόνος εκτέλεσης).
- Αν ο πραγματικός χρόνος εκτέλεσης μιας διεργασίας είναι μεγαλύτερος λόγω της καθυστέρησης του επεξεργαστή τότε το τελευταίο κβάντο θα είναι μεγαλύτερο από το κβάντο που έχουμε δώσει ως όρισμα από το τερματικό.

Procedures_Queue.h (εισάγεται στο αρχείο scheduler.c ως header file)

```
/* header files */
```

<code>#include <stdio.h></code>	κλασική βιβλιοθήκη
<code>#include <stdlib.h></code>	βιβλιοθήκη για fork()
<code>#include <unistd.h></code>	βιβλιοθήκη για execl()
<code>#include <wait.h></code>	για waitpid()
<code>#include <string.h></code>	κλασική βιβλιοθήκη για συναρτήσεις αλφαριθμητικών
<code>#include <sys/time.h></code>	για μέτρηση τρέχοντος χρόνου gettimeofday()
<code>#include <signal.h></code>	για σήματα, π.χ. kill(), sigaction()
<code>#include <math.h></code>	για μαθηματικές συναρτήσεις
<code>#include <fcntl.h></code>	για να μην εκτυπώνονται οι εντολές που εκτελεί το αρχείο Makefile ή το Clean

```
/* global definitions */  
/* global variables and data structures */  
/* definition and implementation of process descriptor and queue(s) */
```

int sigadd = 0; μεταβλητή που παίρνει τιμή 1 μόνο όταν καλείται ο handler. Χρησιμοποιείται στην περίπτωση που μία διεργασία τελειώνει και δεν θα πρέπει να ξαναμπεί στην ουρά.

typedef struct Procedure δομή για την αναπαράσταση κόμβους. Τα πεδία της αποθηκεύουν τον όνομα της διεργασίας την κατάσταση, την προτεραιότητα, τον χρόνο που εκτελείται κάθε φορά, το id της και τον αριθμό που επαναλήψεων που θα χρειαστεί για να ολοκληρωθεί στους δυναμικούς αλγόριθμους.

typedef struct QueueNode αποθήκευση επόμενου και προηγούμενου κόμβου στην ουρά.

typedef struct ProceduresQueue αποθήκευση κεφαλής και ουράς της ουράς.

double get_wtime(void) επιστρέφεται ο τρέχον χρόνος.

QueueNode * getQueueNode(Procedure data, QueueNode * prev) επιστρέφει τον επόμενο κόμβο από αυτόν που δόθηκε ως όρισμα.

int isSize(ProceduresQueue * ref) επιστρέφει το μήκος της ουράς.

int isEmpty(ProceduresQueue * ref) επιστρέφει τιμή 1 αν η ουρά είναι άδεια αλλιώς 0.

ProceduresQueue * getProceduresQueue() επιστρέφει την κεφαλή και την ουρά της ουράς.

void add_element(ProceduresQueue * ref, Procedure data) προσθήκη κόμβου στην ουρά.

`Procedure dequeue(ProceduresQueue * ref)` διαγραφή κόμβου από την ουρά.

`void excecuteProcedure(char* p, double t0)` στην συνάρτηση δίνεται η διεργασία προς εκτέλεση και ο χρόνος έναρξης. Μόλις ο πατέρας μπει στη συνάρτηση δημιουργεί ένα παιδί και μετά το περιμένει να τερματίσει. Η συνάρτηση αυτή χρησιμοποιείται ΜΟΝΟ για τους ΣΤΑΤΙΚΟΥΣ αλγόριθμους χρονοδρομολόγησης. Όσο ο πατέρας περιμένει, το παιδί του παίρνει το path που δόθηκε και με `exec1` εκτελεί το πρόγραμμα του path.

`void execFCFS(ProceduresQueue * ref, double t0, int print_total)` δίνουμε την ουρά στην οποία θα πρέπει να εφαρμόσουμε τους στατικούς αλγόριθμους FCFS, SJF. Η εκτέλεση γίνεται με τα παρακάτω βήματα:

Παίρνουμε τον πρώτο κόμβο της ουράς(κεφαλή) και μπαίνουμε μέσα στο while όσο η ουρά έχει κόμβους.

Ο κόμβος που εισέρχεται διαγράφεται από την ουρά και λαμβάνει κατάσταση RUNNING.

Καλείται η συνάρτηση `excecuteProcedure()` και όταν επιστρέψει από την κλήση της δίνει στον κόμβο κατάσταση EXITED.

Ο τρέχον κόμβος δίνει τη σειρά του στον επόμενο για να εκτελεστεί.

Η επανάληψη θα συνεχιστεί μέχρι να αδειάσει η ουρά δηλαδή να μην υπάρχει άλλη διεργασία προς εκτέλεση.

`int startProcedure(Procedure pr, double t0)` η συνάρτηση είναι παραλλαγή της `excecuteProcedure()` και χρησιμοποιείται στον RR για την αρχικοποίηση των διεργασιών. Αυτή η υλοποίηση χρησιμοποιείται για να ξεκινήσουν να εκτελούνται οι διεργασίες από τον πατέρα ο οποίος δημιουργεί το παιδί και δεν περιμένει όπως πριν να τερματίσει αλλά το σταματά κατευθείαν με σήμα SIG_STOP.

```
/* signal handler(s) */  
void handlectrlc(int sigid)
```

Αποτελεί την συνάρτηση χειρισμού του σήματος SIGCHLD και θέτει την global μεταβλητή `sigadd` σε 1 το οποίο συνεπάγεται ότι η διεργασία έχει ολοκληρωθεί επιτυχώς και μπορεί να αφαιρεθεί από την ουρά.

```
void execQRR(ProceduresQueue * ref, double quant, double t0, int
print_total)
```

Η συνάρτηση `execQRR()` παίρνει μια δομή δεδομένων ουράς με τις αποθηκευμένες διεργασίες που είναι έτοιμες να εκτελεστούν με τον αλγόριθμο Round Robin.

Σε πρώτη φάση αρχικοποιούμε το τρέχον `node` με την κεφαλή της ουράς και ορίζουμε την δομή του `sigaction` για το σήμα `SIGCHLD` ώστε να αγνοείται και κάθε φορά που καλείται ο `handler` για το `SIGCHLD` γίνεται αυτόματο `restart` στις κλήσεις του συστήματος.

Ακολούθως εκτελούμε ένα `while` μέχρι να αδειάσει η ουρά και να τερματίσουν όλες οι διεργασίες. Στο `while` κάνουμε `dequeue` όσο η ουρά έχει κόμβους (`size bigger than zero`).

Εάν το `pid` του τρέχοντος κόμβου είναι ίσο με `-1` αρχικοποιούμε την διεργασία και την εκτελούμε με την συνάρτηση `startProcedure()`. Εάν το `ProId` δεν είναι `-1`, η συνάρτηση μειώνει τη μεταβλητή `ProC` που λειτουργεί ως μετρητής για την κάθε διεργασία.

Στη συνέχεια, η συνάρτηση στέλνει το σήμα `SIGCONT` για σταματημένες διεργασίες και συνεχίζει την εκτέλεση.

Στην περίπτωση που το `ProC` είναι `0`, η διεργασία έχει ολοκληρώθει. Η συνάρτηση αλλάζει τη δομή `sigaction` ώστε να καλεί τον `s_handler` όταν λαμβάνει σήμα `SIGCHLD` επειδή πρόκειται να τερματίσει το παιδί και καλεί την `waitpid` για να περιμένει ο πατέρας το παιδί του να τερματίσει.

Αφού τερματίσει η διεργασία αλλάζουμε τη δομή `sigaction` να αγνοεί το σήμα `SIGCHLD`.

Εάν το `ProC` είναι μεγαλύτερο από `0`, σημαίνει ότι η διαδικασία δεν έχει ολοκληρωθεί. Τότε η συνάρτηση περιμένει χρόνο ίσο με το `κβάντο` και στέλνει σήμα `SIGSTOP`, αυτό το σήμα σταματά την εκτέλεση της διαδικασίας, ορίζει την κατάσταση της διαδικασίας σε `"STOPPED"` και προσθέτει τη διαδικασία πίσω στην ουρά εάν η ουρά δεν είναι κενή ή εάν η διαδικασία δεν έχει ολοκληρωθεί και έχει απομείνει μόνη της στην ουρά.

```
int prioNo(ProceduresQueue * ref)
```

Η συνάρτηση χρησιμοποιείται για τον αλγόριθμο `PRIO` και επιστέφει το πλήθος των διαφορετικών προτεραιοτήτων των διεργασιών.

Στον αλγόριθμο υπάρχει η περίπτωση διεργασιών με την ίδια προτεραιότητα επομένως θα πρέπει να γνωρίζουμε πόσες ουρές πρέπει να φτιάξουμε με διεργασίες που να έχουν την ίδια προτεραιότητα. Κάθε φορά που ο κόμβος μέσα στο `while` έχει ίδια προτεραιότητα με τον προηγούμενο η τιμή του `counter` αυξάνεται. Κάθε κόμβος που εξετάζουμε διαγράφεται από την ουρά και μόλις

ελεγχθεί ξανά εισάγεται, έτσι η ουρά διατηρείται. Η τιμή των διαφορετικών προτεραιοτήτων επιστρέφεται με την κλήση της συνάρτησης.

scheduler.c

- Αρχικά, εισάγουμε ως βιβλιοθήκη το αρχείο `Procedures_Queue.h` το οποίο περιέχει τις προαναφερθείσες χρήσιμες συναρτήσεις.
- Ύστερα, ορίζουμε τις συναρτήσεις `FCFS`, `SJF`, `RR` και `PRIO`.
- Και υλοποιούμε την συνάρτηση `main`.

```
/* implementation of the scheduling policies, etc. batch(), rr() etc. */
```

Η συνάρτηση `FCFS` υλοποιείται ως εξής:

```
void FCFS(int count, char* test_file)
```

Δημιουργούμε μια ουρά κόμβων που αντιπροσωπεύουν διεργασίες και ύστερα διαπερνάμε τις γραμμές του αρχείου εισόδου και για κάθε γραμμή δημιουργούμε έναν κόμβο στον οποίο βάζουμε μόνο το μονοπάτι του αντίστοιχου εκτελέσιμου αρχείου και την κατάστασή του (`READY`). Τα υπόλοιπα πεδία του κόμβου τα αφήνουμε κενά επειδή δεν τα χρησιμοποιούμε σε αυτήν την περίπτωση. Ύστερα βάζουμε τον κόμβο στην ουρά και συνεχίζουμε με την επόμενη γραμμή του αρχείου. Τέλος, εκτελούμε τις διεργασίες που υπάρχουν στην ουρά.

Η λογική πίσω από τον `SJF` είναι η εξής:

```
void SJF(int count, char* test_file)
```

Παίρνουμε από το αρχείο εισόδου τα ονόματα και τους χρόνους εκτέλεσης των εκτελέσιμων αρχείων `work` και τα αποθηκεύουμε σε πίνακες.

Ύστερα, παίρνουμε από τους πίνακες την διεργασία με τον μικρότερο χρόνο εκτέλεσης, δημιουργούμε τον αντίστοιχο κόμβο και τον βάζουμε στην ουρά. Αυτό επαναλαμβάνεται μέχρι να διαβάσουμε όλες τις γραμμές του αρχείου. Τελειώνοντας, εκτελούμε τις διεργασίες της ουράς όπως στον `FCFS`.

Επιλέξαμε να ταξινομούμε τις διεργασίες `PRIN` τις βάλουμε στην ουρά για να βελτιστοποιήσουμε τον χρόνο ταξινόμησης.

Η συνάρτηση RR υλοποιείται ως εξής:

```
void RR(int count, char* test_file, int quant)
```

Δημιουργούμε μια ουρά και υστέρα γεμίζουμε την ουρά με κόμβους με δεδομένα που προέρχονται από την σειριακή προσπέλαση των γραμμών του αρχείου εισόδου. Μετά, καλούμε την συνάρτηση execQRR η οποία αναλαμβάνει την εκτέλεση της ουράς χρησιμοποιώντας τον αλγόριθμο Round Robin.

Η λογική πίσω από τον PRIO είναι η εξής:

```
void PRIO(int count, char* test_file, int quant)
```

Παίρνουμε από το αρχείο εισόδου τα ονόματα, τους χρόνους εκτέλεσης και τις προτεραιότητες των εκτελέσιμων αρχείων work και τα αποθηκεύουμε σε πίνακες.

Ακολούθως, παίρνουμε από τους πίνακες την διεργασία με την μικρότερη προτεραιότητα, δημιουργούμε τον αντίστοιχο κόμβο και τον βάζουμε στην ουρά. Αυτό επαναλαμβάνεται μέχρι να διαβάσουμε όλες τις γραμμές του αρχείου.

Έπειτα, βρίσκουμε το πλήθος των διακριτών τιμών προτεραιοτήτων prioC και δηλώνουμε έναν πίνακα μεγέθους prioC με ουρές στον οποίο θα τοποθετήσουμε ουρές οι οποίες θα έχουν κόμβους με ίδιες προτεραιότητες.

Υστερα, γεμίζουμε τον πίνακα, και προελαύνουμε με την σειρά τα κελιά του και η ουρά που έχουμε αποθηκεύσει σε ένα κελί έχει μόνο ένα κόμβο τον εκτελούμε με FCFS αλλιώς με RR.

```
int lines_count(char * test_file)
```

Η συνάρτηση lines_count επιστρέφει το πλήθος των γραμμών ενός αρχείου.

```
int main(int argc, char *argv[])
```

Η συνάρτηση main αρχικά εκτελεί το αρχείο Makefile και στην συνέχεια δηλώνει και αρχικοποιεί μεταβλητές σχετικά με την πολιτική δρομολόγησης, το κβάντο και όνομα του αρχείου εισόδου καθώς και το πλήθος των γραμμών του. Υστερα, καλεί την κατάλληλη συνάρτηση δρομολόγησης σύμφωνα με την πολιτική που έχουμε δώσει από το τερματικό. Τέλος, καλούμε το αρχείο Clean το οποίο διαγράφει τα εκτελέσιμα αρχεία work που δημιουργήσαμε στην αρχή της συνάρτησης.

Αποτελέσματα εκτέλεσης run.sh

----> BATCH <---

PID 3487 - CMD: work7

process 3487 begins

process 3487 ends

Elapsed time = 7.374 secs

Workload time = 7.374 secs

PID 3488 - CMD: work7

process 3488 begins

process 3488 ends

Elapsed time = 7.371 secs

Workload time = 14.745 secs

PID 3489 - CMD: work7

process 3489 begins

process 3489 ends

Elapsed time = 7.349 secs

Workload time = 22.094 secs

PID 3490 - CMD: work7

process 3490 begins

process 3490 ends

Elapsed time = 7.325 secs

Workload time = 29.419 secs

PID 3491 - CMD: work7

process 3491 begins

process 3491 ends

Elapsed time = 7.412 secs

Workload time = 36.831 secs

WORKLOAD TIME: 36.831 seconds

----> SJF <---

PID 3531 - CMD: work7

process 3531 begins


```
process 3531 ends
                                Elapsed time = 7.331 secs
                                Workload time = 7.331 secs

PID 3532 - CMD: work7
process 3532 begins
process 3532 ends
                                Elapsed time = 7.354 secs
                                Workload time = 14.685 secs

PID 3533 - CMD: work7
process 3533 begins
process 3533 ends
                                Elapsed time = 7.337 secs
                                Workload time = 22.022 secs

PID 3534 - CMD: work7
process 3534 begins
process 3534 ends
                                Elapsed time = 7.367 secs
                                Workload time = 29.389 secs

PID 3536 - CMD: work7
process 3536 begins
process 3536 ends
                                Elapsed time = 7.355 secs
                                Workload time = 36.743 secs
```

WORKLOAD TIME: 36.743 seconds

----> RR <----

```
process 3576 begins
PID 3576 - CMD: ../work/work7
                                Elapsed time = 1.000 secs
                                Workload time = 1.000 secs

process 3577 begins
PID 3577 - CMD: ../work/work7
```

```

        Elapsed time = 1.000 secs
        Workload time = 2.000 secs

process 3578 begins
PID 3578 - CMD: ../work/work7
        Elapsed time = 1.000 secs
        Workload time = 3.000 secs

process 3579 begins
PID 3579 - CMD: ../work/work7
        Elapsed time = 1.000 secs
        Workload time = 4.001 secs

process 3580 begins
PID 3580 - CMD: ../work/work7
        Elapsed time = 1.000 secs
        Workload time = 5.001 secs

PID 3576 - CMD: ../work/work7
        Elapsed time = 2.000 secs
        Workload time = 6.001 secs

PID 3577 - CMD: ../work/work7
        Elapsed time = 2.000 secs
        Workload time = 7.001 secs

PID 3578 - CMD: ../work/work7
        Elapsed time = 2.000 secs
        Workload time = 8.001 secs

PID 3579 - CMD: ../work/work7
        Elapsed time = 2.000 secs
        Workload time = 9.001 secs

PID 3580 - CMD: ../work/work7
        Elapsed time = 2.000 secs
        Workload time = 10.001 secs

PID 3576 - CMD: ../work/work7
        Elapsed time = 3.000 secs
        Workload time = 11.001 secs

PID 3577 - CMD: ../work/work7
        Elapsed time = 3.000 secs
        Workload time = 12.001 secs

PID 3578 - CMD: ../work/work7
        Elapsed time = 3.000 secs
        Workload time = 13.001 secs
```

```
PID 3579 - CMD: ../work/work7
    Elapsed time = 3.000 secs
    Workload time = 14.002 secs

PID 3580 - CMD: ../work/work7
    Elapsed time = 3.000 secs
    Workload time = 15.002 secs

PID 3576 - CMD: ../work/work7
    Elapsed time = 4.000 secs
    Workload time = 16.002 secs

PID 3577 - CMD: ../work/work7
    Elapsed time = 4.000 secs
    Workload time = 17.002 secs

PID 3578 - CMD: ../work/work7
    Elapsed time = 4.000 secs
    Workload time = 18.002 secs

PID 3579 - CMD: ../work/work7
    Elapsed time = 4.000 secs
    Workload time = 19.002 secs

PID 3580 - CMD: ../work/work7
    Elapsed time = 4.000 secs
    Workload time = 20.002 secs

PID 3576 - CMD: ../work/work7
    Elapsed time = 5.000 secs
    Workload time = 21.002 secs

PID 3577 - CMD: ../work/work7
    Elapsed time = 5.000 secs
    Workload time = 22.002 secs

PID 3578 - CMD: ../work/work7
    Elapsed time = 5.000 secs
    Workload time = 23.002 secs

PID 3579 - CMD: ../work/work7
    Elapsed time = 5.000 secs
    Workload time = 24.002 secs

PID 3580 - CMD: ../work/work7
    Elapsed time = 5.000 secs
    Workload time = 25.002 secs

PID 3576 - CMD: ../work/work7
    Elapsed time = 6.000 secs
```

```

                                Workload time = 26.003 secs
PID 3577 - CMD: ../work/work7
                                Elapsed time = 6.000 secs
                                Workload time = 27.003 secs
PID 3578 - CMD: ../work/work7
                                Elapsed time = 6.000 secs
                                Workload time = 28.003 secs
PID 3579 - CMD: ../work/work7
                                Elapsed time = 6.000 secs
                                Workload time = 29.003 secs
PID 3580 - CMD: ../work/work7
                                Elapsed time = 6.000 secs
                                Workload time = 30.003 secs
process 3576 ends
PID 3576 - CMD: ../work/work7
                                Elapsed time = 7.379 secs
                                Workload time = 31.382 secs
process 3577 ends
PID 3577 - CMD: ../work/work7
                                Elapsed time = 7.450 secs
                                Workload time = 32.832 secs
process 3578 ends
PID 3578 - CMD: ../work/work7
                                Elapsed time = 7.397 secs
                                Workload time = 34.229 secs
process 3579 ends
PID 3579 - CMD: ../work/work7
                                Elapsed time = 7.356 secs
                                Workload time = 35.585 secs
process 3580 ends
PID 3580 - CMD: ../work/work7
                                Elapsed time = 7.349 secs
                                Workload time = 36.934 secs
```

WORKLOAD TIME: 36.934 seconds

---> BATCH <---

```
PID 3620 - CMD: work7
process 3620 begins
process 3620 ends

Elapsed time = 7.343 secs
Workload time = 7.343 secs

PID 3621 - CMD: work6
process 3621 begins
process 3621 ends

Elapsed time = 6.273 secs
Workload time = 13.616 secs

PID 3622 - CMD: work5
process 3622 begins
process 3622 ends

Elapsed time = 5.256 secs
Workload time = 18.873 secs

PID 3623 - CMD: work4
process 3623 begins
process 3623 ends

Elapsed time = 4.246 secs
Workload time = 23.118 secs

PID 3624 - CMD: work3
process 3624 begins
process 3624 ends

Elapsed time = 3.191 secs
Workload time = 26.309 secs

PID 3625 - CMD: work2
process 3625 begins
process 3625 ends

Elapsed time = 2.106 secs
Workload time = 28.416 secs

PID 3626 - CMD: work1
process 3626 begins
process 3626 ends

Elapsed time = 1.079 secs
Workload time = 29.495 secs
```

WORKLOAD TIME: 29.495 seconds

---> SJF <---

PID 3666 - CMD: work1

process 3666 begins

process 3666 ends

Elapsed time = 1.063 secs

Workload time = 1.063 secs

PID 3667 - CMD: work2

process 3667 begins

process 3667 ends

Elapsed time = 2.140 secs

Workload time = 3.203 secs

PID 3668 - CMD: work3

process 3668 begins

process 3668 ends

Elapsed time = 3.196 secs

Workload time = 6.399 secs

PID 3669 - CMD: work4

process 3669 begins

process 3669 ends

Elapsed time = 4.206 secs

Workload time = 10.604 secs

PID 3670 - CMD: work5

process 3670 begins

process 3670 ends

Elapsed time = 5.386 secs

Workload time = 15.990 secs

PID 3671 - CMD: work6

process 3671 begins

process 3671 ends

Elapsed time = 6.274 secs

Workload time = 22.265 secs

PID 3672 - CMD: work7

process 3672 begins

process 3672 ends

Elapsed time = 7.408 secs

Workload time = 29.673 secs

WORKLOAD TIME: 29.673 seconds

---> RR <---

process 3712 begins

PID 3712 - CMD: ../work/work7

Elapsed time = 1.000 secs

Workload time = 1.000 secs

process 3713 begins

PID 3713 - CMD: ../work/work6

Elapsed time = 1.000 secs

Workload time = 2.001 secs

process 3714 begins

PID 3714 - CMD: ../work/work5

Elapsed time = 1.000 secs

Workload time = 3.001 secs

process 3715 begins

PID 3715 - CMD: ../work/work4

Elapsed time = 1.000 secs

Workload time = 4.001 secs

process 3716 begins

PID 3716 - CMD: ../work/work3

Elapsed time = 1.000 secs

Workload time = 5.001 secs

process 3717 begins

PID 3717 - CMD: ../work/work2

Elapsed time = 1.000 secs

Workload time = 6.001 secs

process 3718 begins

process 3718 ends

PID 3718 - CMD: ../work/work1

Elapsed time = 1.043 secs

Workload time = 7.044 secs

PID 3712 - CMD: ../work/work7

Elapsed time = 2.000 secs

Workload time = 8.044 secs

PID 3713 - CMD: ../work/work6

```

        Elapsed time = 2.000 secs
        Workload time = 9.044 secs
PID 3714 - CMD: ../work/work5
        Elapsed time = 2.000 secs
        Workload time = 10.044 secs
PID 3715 - CMD: ../work/work4
        Elapsed time = 2.000 secs
        Workload time = 11.044 secs
PID 3716 - CMD: ../work/work3
        Elapsed time = 2.000 secs
        Workload time = 12.044 secs
process 3717 ends
PID 3717 - CMD: ../work/work2
        Elapsed time = 2.102 secs
        Workload time = 13.146 secs
PID 3712 - CMD: ../work/work7
        Elapsed time = 3.000 secs
        Workload time = 14.146 secs
PID 3713 - CMD: ../work/work6
        Elapsed time = 3.000 secs
        Workload time = 15.146 secs
PID 3714 - CMD: ../work/work5
        Elapsed time = 3.000 secs
        Workload time = 16.146 secs
PID 3715 - CMD: ../work/work4
        Elapsed time = 3.000 secs
        Workload time = 17.146 secs
process 3716 ends
PID 3716 - CMD: ../work/work3
        Elapsed time = 3.140 secs
        Workload time = 18.286 secs
PID 3712 - CMD: ../work/work7
        Elapsed time = 4.000 secs
        Workload time = 19.286 secs
PID 3713 - CMD: ../work/work6
        Elapsed time = 4.000 secs
        Workload time = 20.286 secs
PID 3714 - CMD: ../work/work5
```



```

                Elapsed time = 4.000 secs
                Workload time = 21.286 secs

process 3715 ends

PID 3715 - CMD: ../work/work4

                Elapsed time = 4.200 secs
                Workload time = 22.486 secs

PID 3712 - CMD: ../work/work7

                Elapsed time = 5.000 secs
                Workload time = 23.486 secs

PID 3713 - CMD: ../work/work6

                Elapsed time = 5.000 secs
                Workload time = 24.486 secs

process 3714 ends

PID 3714 - CMD: ../work/work5

                Elapsed time = 5.279 secs
                Workload time = 25.765 secs

PID 3712 - CMD: ../work/work7

                Elapsed time = 6.000 secs
                Workload time = 26.770 secs

process 3713 ends

PID 3713 - CMD: ../work/work6

                Elapsed time = 6.314 secs
                Workload time = 28.084 secs

process 3712 ends

PID 3712 - CMD: ../work/work7

                Elapsed time = 7.504 secs
                Workload time = 29.589 secs
```

WORKLOAD TIME: 29.589 seconds

---> PRIO <---

```

process 3758 begins

PID 3758 - CMD: ../work/work4

                Elapsed time = 0.500 secs
                Workload time = 0.500 secs

process 3759 begins

PID 3759 - CMD: ../work/work6
```

```

        Elapsed time = 0.500 secs
        Workload time = 1.000 secs
PID 3758 - CMD: ../work/work4
        Elapsed time = 1.000 secs
        Workload time = 1.500 secs
PID 3759 - CMD: ../work/work6
        Elapsed time = 1.000 secs
        Workload time = 2.000 secs
PID 3758 - CMD: ../work/work4
        Elapsed time = 1.500 secs
        Workload time = 2.500 secs
PID 3759 - CMD: ../work/work6
        Elapsed time = 1.500 secs
        Workload time = 3.000 secs
PID 3758 - CMD: ../work/work4
        Elapsed time = 2.000 secs
        Workload time = 3.500 secs
PID 3759 - CMD: ../work/work6
        Elapsed time = 2.000 secs
        Workload time = 4.000 secs
PID 3758 - CMD: ../work/work4
        Elapsed time = 2.500 secs
        Workload time = 4.501 secs
PID 3759 - CMD: ../work/work6
        Elapsed time = 2.500 secs
        Workload time = 5.001 secs
PID 3758 - CMD: ../work/work4
        Elapsed time = 3.000 secs
        Workload time = 5.501 secs
PID 3759 - CMD: ../work/work6
        Elapsed time = 3.000 secs
        Workload time = 6.001 secs
PID 3758 - CMD: ../work/work4
        Elapsed time = 3.500 secs
        Workload time = 6.501 secs
PID 3759 - CMD: ../work/work6
        Elapsed time = 3.500 secs
        Workload time = 7.001 secs
```

```
process 3758 ends
PID 3758 - CMD: ../work/work4
    Elapsed time = 4.266 secs
    Workload time = 7.767 secs
PID 3759 - CMD: ../work/work6
    Elapsed time = 4.000 secs
    Workload time = 8.267 secs
PID 3759 - CMD: ../work/work6
    Elapsed time = 4.500 secs
    Workload time = 8.768 secs
PID 3759 - CMD: ../work/work6
    Elapsed time = 5.000 secs
    Workload time = 9.268 secs
PID 3759 - CMD: ../work/work6
    Elapsed time = 5.500 secs
    Workload time = 9.768 secs
process 3759 ends
PID 3759 - CMD: ../work/work6
    Elapsed time = 6.374 secs
    Workload time = 10.642 secs
process 3763 begins
PID 3763 - CMD: ../work/work5
    Elapsed time = 0.500 secs
    Workload time = 11.142 secs
process 3764 begins
PID 3764 - CMD: ../work/work7
    Elapsed time = 0.500 secs
    Workload time = 11.643 secs
PID 3763 - CMD: ../work/work5
    Elapsed time = 1.000 secs
    Workload time = 12.143 secs
PID 3764 - CMD: ../work/work7
    Elapsed time = 1.000 secs
    Workload time = 12.643 secs
PID 3763 - CMD: ../work/work5
    Elapsed time = 1.500 secs
    Workload time = 13.143 secs
PID 3764 - CMD: ../work/work7
```

```

        Elapsed time = 1.500 secs
        Workload time = 13.643 secs
PID 3763 - CMD: ../work/work5
        Elapsed time = 2.000 secs
        Workload time = 14.143 secs
PID 3764 - CMD: ../work/work7
        Elapsed time = 2.000 secs
        Workload time = 14.643 secs
PID 3763 - CMD: ../work/work5
        Elapsed time = 2.500 secs
        Workload time = 15.143 secs
PID 3764 - CMD: ../work/work7
        Elapsed time = 2.500 secs
        Workload time = 15.643 secs
PID 3763 - CMD: ../work/work5
        Elapsed time = 3.000 secs
        Workload time = 16.143 secs
PID 3764 - CMD: ../work/work7
        Elapsed time = 3.000 secs
        Workload time = 16.643 secs
PID 3763 - CMD: ../work/work5
        Elapsed time = 3.500 secs
        Workload time = 17.143 secs
PID 3764 - CMD: ../work/work7
        Elapsed time = 3.500 secs
        Workload time = 17.643 secs
PID 3763 - CMD: ../work/work5
        Elapsed time = 4.000 secs
        Workload time = 18.143 secs
PID 3764 - CMD: ../work/work7
        Elapsed time = 4.000 secs
        Workload time = 18.643 secs
PID 3763 - CMD: ../work/work5
        Elapsed time = 4.500 secs
        Workload time = 19.143 secs
PID 3764 - CMD: ../work/work7
        Elapsed time = 4.500 secs
        Workload time = 19.643 secs
```

```
process 3763 ends
PID 3763 - CMD: ../work/work5
    Elapsed time = 5.222 secs
    Workload time = 20.365 secs
PID 3764 - CMD: ../work/work7
    Elapsed time = 5.000 secs
    Workload time = 20.865 secs
PID 3764 - CMD: ../work/work7
    Elapsed time = 5.500 secs
    Workload time = 21.365 secs
PID 3764 - CMD: ../work/work7
    Elapsed time = 6.000 secs
    Workload time = 21.866 secs
PID 3764 - CMD: ../work/work7
    Elapsed time = 6.500 secs
    Workload time = 22.366 secs
process 3764 ends
PID 3764 - CMD: ../work/work7
    Elapsed time = 7.348 secs
    Workload time = 23.214 secs
```

WORKLOAD TIME: 23.214 seconds