

Λειτουργικά συστήματα, Εργασία 1, Περιττοί ΑΜ

ΠΑΝΑΓΙΩΤΗΣ ΠΕΤΡΑΚΟΠΟΥΛΟΣ - 1115 2021 00155

November 19, 2023

1 Εισαγωγή

Το project είναι υλοποιημένο σε **C++/17**. Περιέχει τις υλοποιήσεις των δύο διεργασιών που ζητήθηκαν, ένα common φακέλο που περιέχει classes που χρησιμοποιούνται και από τις δύο διεργασίες, Unit Tests για τα common objects και ένα make file.

2 Common folder

Το common folder περιέχει τα εξής classes:

(*Resource Wrapper: A class that handles the lifecycle of a system resource and provides an abstract interface for the resource's methods)

2.1 Thread

Είναι ένα απλό resource wrapper του pthread_t που παρέχει τις επιλογές Start και Join.

2.2 Semaphore

Είναι ένα απλό resource wrapper του sem_t που παρέχει τις επιλογές Post, Wait και Value.

2.3 SharedMemory

Είναι ένα resource wrapper του shm. Στο Initialize δημιουργεί το shared memory segment, το κάνει ftruncate και mmap, ενώ στο open απλά mmap. παρέχει τις μεθόδους read και write, καθώς και έναν subscript operator που μπορεί να χρησιμοποιηθεί για πιο άμεση πρόσβαση στο memory segment.

2.4 Constants

Είναι ένα class που περιέχει κάποια constants που χρησιμοποιούνται απο τις διεργασίες καθώς και απο κάποια resource wrappers έτσι ώστε να αποφεύγεται η χρήση "μαγικών σταθερών" η καρφωτών τιμών.

2.5 Buffer

Αποτελεί το Base Class για άλλα buffers και δεν θα έπρεπε να χρησιμοποιείται απο μόνο του. Παρέχει τις κατάλληλες μεθόδους για το άνοιγμα η την αρχικοποίηση ενός buffer που μπορεί να χρησιμοποιηθεί απο διαφορετικές διεργασίες. Στην Αρχικοποίηση η το άνοιγμα του αρχικοποιεί/ανοίγει 3 σεμαφόρους για την διαχείριση το buffer και ένα αντικείμενο κοινής μνήμης

2.6 WriteOnlyBuffer

Είναι subclass του Buffer. Χρησιμοποιεί τους σεμαφόρους και τη κοινόχρηστη μνήμη για να γράφει στον buffer όταν υπάρχει ελεύθερος χώρος και όταν δεν χρησιμοποιείται ο buffer απο κάποιον άλλο. Όταν του δίνεται ένα μήνυμα τύπου `std::string`, το σπάει σε "πακέτα" 15 χαρακτήρων και τα στέλνει ένα ένα. Γράφει ένα πακέτο και στη συνέχεια περιμένει μέχρι αυτό να διαβαστεί. Μόλις αδειάσει ο buffer, γράφει και άλλο ένα πακέτο και αυτό συνεχίζεται μέχρις ότου να γραφούν όλα τα πακέτα του μηνύματος. Το μήνυμα τελειώνει πάντα σε έναν null termination character.

2.7 ReadOnlyBuffer

Είναι subclass του Buffer. Χρησιμοποιεί τους σεμαφόρους και τη κοινόχρηστη μνήμη για να διαβάσει απο τον buffer όταν υπάρχουν δεδομένα και όταν δεν χρησιμοποιείται ο buffer απο κάποιον άλλο. Διαβάζει συνεχώς απο τον buffer όσο υπάρχουν δεδομένα, και αν λαμβάνει περισσότερα απο 1 πακέτα τα κολλάει μαζί έτσι ώστε να πάρει το ολόκληρο μήνυμα ο τελικός χρήστης. Σταματάει να διαβάζει μόλις βρει έναν null termination character που σημαίνει πως το μήνυμα τελείωσε.

2.8 SenderThread

Το sender thread είναι απλά ένα function με το εξής signature:

```
void* SenderThread(void* arg);
```

το signature του είναι τέτοιο ώστε να μπορεί να κληθεί απο pthreads. Στη συγκεκριμένη περίπτωση περιμένει πως το arg είναι τύπου *WriteOnlyBuffer**. Περιμένει για είσοδο απο το πληκτρολόγιο και μόλις διαβάσει κάτι το γράφει στο buffer με την χρήση το *WriteOnlyBuffer*. Όταν διαβάσει τη γραμματοσειρά "#BYE#" και την συνάρτηση `exit()` για να σταματήσει η εκτέλεση του προγράμματος.

2.9 ReceiverThread

Το sender thread είναι απλά ένα function με το εξής signature:

```
void* ReceiverThread(void* arg);
```

το signature του είναι τέτοιο ώστε να μπορεί να κληθεί από pthreads. Στη συγκεκριμένη περίπτωση περιμένει πως το arg είναι τύπου *ReadOnlyBuffer**. Περιμένει να υπάρξουν δεδομένα στον buffer και μόλις βρεί τα εμφανίζει στο terminal του προγράμματος.

3 Tests

Υπάρχουν 4 tests σε αυτόν τον φάκελο τα οποία αρχικοποιούν τα αντικείμενα τα common directory και τρέχουν κάποια βασικά operations πάνω σε αυτά έτσι ώστε να διασφαλιστεί πως έχουν σωστή συμπεριφορά.

4 Process A

Αρχικοποιεί δύο Buffer αντικείμενα. Έναν *ReadOnlyBuffer* και έναν *WriteOnlyBuffer*. Στη συνέχεια δημιουργεί 2 threads. Ένα sender thread και ένα receiver thread. Τους περνάει ως παραμέτρους το αντίστοιχο buffer και function για κάθε μία και τα ξεκινάει. Στη συνέχεια τα κάνει Join και τα περιμένει για να ολοκληρωθεί η εκτέλεση της.

5 Process B

Ανοίγει δύο Buffer αντικείμενα αφού θα έχουν αρχικοποιηθεί ήδη από το Process A. Έναν *ReadOnlyBuffer* και έναν *WriteOnlyBuffer*. Στη συνέχεια δημιουργεί 2 threads. Ένα sender thread και ένα receiver thread. Τους περνάει ως παραμέτρους το αντίστοιχο buffer και function για κάθε μία και τα ξεκινάει. Στη συνέχεια τα κάνει Join και τα περιμένει για να ολοκληρωθεί η εκτέλεση της.

6 Makefile

Το Makefile έχει τις εξής επιλογές:

6.1 Common

Κάνει build σε object files όλα τα classes του common folder έτσι ώστε να χρησιμοποιηθούν στις διαδικασίες A & B.

6.2 Tests

Κάνει build τα Unit Tests του Common folder και τα τρέχει έτσι ώστε να εξασφαλιστεί πως δεν υπάρχουν λάθη.

6.3 A

Κάνει build την διαδικασία A και κάνει link όλα τα απαραίτητα objects απο το common directory.

6.4 B

Κάνει build την διαδικασία B και κάνει link όλα τα απαραίτητα objects απο το common directory.

6.5 Clean

Διαγράφει όλα files που μπορεί να έχουν δημιουργηθεί απο το Makefile στα τα bin και build directories

6.6 Proj

Χτίζει όλο το project απο την αρχή. Καλεί στην εξής σειρά το Makefile:

1. Clean
2. Common
3. A
4. B
5. Tests