Πανεπιστήμιο Δυτικής Μακεδονίας
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Λειτουργικά Συστήματα

Ενότητα 11: Αμοιβαίος αποκλεισμός με σημαφόρους SystemV και POSIX . Διαχείριση αρχείων. Προσομοιωτής μνήμης MOSS-memory.

Επιβλέπων: Δασυγένης Μηνάς
Παυλίδου Ελένη

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Λειτουργικών Συστημάτων
http://arch.ece.uowm.gr/courses/os/

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σημαφόροι SystemV, POSIX (1/2)

- Στα συστήματα UNIX/LINUX έχουν επικρατήσει δυο είδη σημαφόρων: Οι σημαφόροι τύπου SystemV και οι σημαφόροι τύπου Posix.

- Στο SystemV χρησιμοποιούνται 3 συναρτήσεις για τη διαχείριση των σημαφόρων και συγκεκριμένα οι `semget()`, `semop()`, `semctl()`.

- Το POSIX αποτελεί μια οικογένεια προτύπων και είναι μια συντομογραφία του όρου «Φορητή διεπαφή λειτουργικών συστημάτων» (Portable Operating System Interface).

Σημαφόροι SystemV, POSIX (2/2)

- Το POSIX προσφέρει ένα σύνολο προτύπων παροχής κριτηρίων συμμόρφωσης για υπηρεσίες λειτουργικών συστημάτων και είναι σχεδιασμένο να επιτρέπει σε προγράμματα εφαρμογών να γράφουν εφαρμογές που μπορούν εύκολα να μεταφέρονται μεταξύ διαφορετικών λειτουργικών συστημάτων.
  Προσδιορίζει επίσης έναν καθορισμένο τρόπο για μια εφαρμογή να αλληλεπιδρά με το λειτουργικό σύστημα.

Η συνάρτηση semget() (1/2)

- Όπως και οι ουρές μηνυμάτων, ο σηματοφορέας πρέπει να αρχικοποιηθεί με τη συνάρτηση `semget()` . Η συνάρτηση `semget()` προτυποποιείται ως:

```
int semget(key_t key, int nsems, int
semflg);
```

- Αν η κλήση επιτύχει, επιστρέφει το ID του σηματοφορέα (semid).

Η συνάρτηση semget() (2/2)

- Το όρισμα `key` είναι ίσο με το IPC_PRIVATE.
- Το όρισμα `nsems` καθορίζει τον αριθμό των στοιχείων στον πίνακα σηματοφορέων. Η κλήση αποτυγχάνει όταν το `nsems` είναι μεγαλύτερο από τον αριθμό των διαθέσιμων στοιχείων. Όταν ο αριθμός των διαθέσιμων στοιχείων δεν είναι γνωστός, η τιμή `0` σε αυτό το όρισμα εξασφαλίζει την επιτυχία της κλήσης.
- Το όρισμα `semflg` καθορίζει τα αρχικά δικαιώματα πρόσβασης και δημιουργίας.

Η συνάρτηση semop() (1/2)

- Η συνάρτηση `semop()` εκτελεί τις λειτουργίες των σηματοφορέων. Δηλώνεται ως:

```
int semop(int semid, struct sembuf *sops,
size_t nsops);
```

- Το όρισμα `semid` πρέπει να αναφέρεται σε ID υπάρχοντος σηματοφορέα (ή πίνακα σηματοφορέων).

Η συνάρτηση semop() (2/2)

- Το όρισμα `sops` είναι δείκτης σε έναν πίνακα δομών τύπου sembuf, όπου η κάθε μια περιέχει τις παρακάτω πληροφορίες για τη λειτουργία ενός σηματοφορέα:

- Αριθμός σηματοφορέα.
- Λειτουργία προς εκτέλεση.
- Σημαίες ελέγχου, αν υπάρχουν.

❖ Η δομή `sembuf` ορίζεται στη βιβλιοθήκη `sys/sem.h`.

Η συνάρτηση semct() (1/5)

- Η συνάρτηση `semctl()` τροποποιεί τα δικαιώματα πρόσβασης και άλλα χαρακτηριστικά του σηματοφορέα. Δηλώνεται ως εξής:

```
int semctl(int semid, int semnum, int cmd,
union semun arg);
```

- Το όρισμα `semid` πρέπει να αναφέρεται σε ID υπάρχοντος σηματοφορέα. Η τιμή του `semnum` ορίζει τη θέση ενός σηματοφορέα σε σχετικό πίνακα.

Η συνάρτηση semct() (2/5)

- Το όρισμα `cmd` λαμβάνει μια από τις παρακάτω τιμές ελέγχου:

- **GETVAL** -- Επιστρέφει την τιμή απλού σηματοφορέα. Σε αυτήν την περίπτωση η τιμή βρίσκεται στον ακέραιο `arg.val`.

- **SETVAL** -- Θέτει την τιμή απλού σηματοφορέα. Σε αυτήν την περίπτωση η τιμή βρίσκεται στον ακέραιο `arg.val`.

- **GETPID**-- Επιστρέφει PID της διεργασίας που προσπέλασε τελευταία το σηματοφορέα ή τον πίνακα.

Η συνάρτηση semct() (3/5)

- **GETNCNT**-- Επιστρέφει τον αριθμό των διεργασιών που αναμένουν αύξηση της τιμής του σηματοφορέα.
- **GETZCNT**-- Επιστρέφει τον αριθμό των διεργασιών που αναμένουν να λάβει ο σηματοφορέας τιμή μηδέν.
- **GETALL**-- Επιστρέφει τις τιμές όλων των σηματοφορέων (σε πίνακα). Σε αυτή τη περίπτωση οι τιμές βρίσκονται μέσω του `arg.array`, που είναι δείκτης σε πίνακα unsigned short.

Η συνάρτηση semct() (4/5)

- **SETALL**-- Θέτει τις τιμές όλων των σηματοφορέων (σε πίνακα). Σε αυτή τη περίπτωση οι τιμές βρίσκονται μέσω του `arg.array`, που είναι δείκτης σε πίνακα `unsigned short`.

- **IPC_STAT**-- Επιστρέφει πληροφορίες για την κατάσταση. Σε αυτήν την περίπτωση η πληροφορία βρίσκεται μέσω του `arg.buf`, που είναι δείκτης σε δομή τύπου `semid_ds`.

- **IPC_SET**-- Θέτει πληροφορίες για την κατάσταση. Σε αυτήν την περίπτωση η πληροφορία βρίσκεται μέσω του `arg.buf`, που είναι δείκτης σε δομή τύπου `semid_ds`.

Η συνάρτηση semct() (5/5)

- **IPC_RMID** -- Διαγράφει το σύνολο των σηματοφορέων.

❖ Τα όρισμα ελέγχου απαιτεί και ανάλογα δικαιώματα της καλούσας διεργασίας. Για τις επιλογές IPC_SET ή IPC_RMID απαιτείται δικαίωμα διαχειριστή ή ιδιοκτήτη. Για τις υπόλοιπες επιλογές απαιτούνται δικαιώματα εγγραφής και ανάγνωσης.
Το τέταρτο όρισμα `union semun arg` είναι προαιρετικό και χρησιμοποιείται μόνο αν το απαιτεί το όρισμα ελέγχου.

POSIX (1/6)

- Οι σηματοφορείς POSIX είναι πολύ ελαφρύτεροι από τους σηματοφορείς του System V. Ο σηματοφορέας POSIX είναι πάντα απλός, όχι πίνακας έως 25 σηματοφορέων όπως το System V.

- Οι πιο σημαντικές συναρτήσεις για τους σημαφόρους του POSIX είναι η `sem_init()`, η `sem_wait()`, η `sem_post()` και η `sem_destroy()`.

POSIX (2/6)

- Οι συναρτήσεις χειρισμού των σηματοφορέων POSIX είναι:

- `sem_open()` -- Προσπελαύνει, και προαιρετικά δημιουργεί, ένα ονοματισμένο σηματοφορέα

- `sem_init()` -- Αρχικοποιεί ένα σηματοφορέα (εσωτερικά στο πρόγραμμα που καλεί τη συνάρτηση, δηλαδή όχι ονοματισμένο).

POSIX (3/6)

- `sem_close()` -- Τερματίζει την προσπέλαση σε ανοικτό σηματοφορέα.
- `sem_unlink()` -- Τερματίζει την προσπέλαση σε ανοικτό σηματοφορέα, και καταργεί το σηματοφορέα όταν όλες οι διεργασίες τερματίσουν τη προσπέλασή τους.
- `sem_destroy()` -- Τερματίζει ένα σηματοφορέα (εσωτερικά στο πρόγραμμα που καλεί τη συνάρτηση, δηλαδή όχι ονοματισμένο).

POSIX (4/6)

- `sem_getvalue()` -- Αντιγράφει την τιμή του σηματοφορέα σε δεδομένο ακέραιο.

- `sem_wait(), sem_trywait()` -- Αναστέλλει τη διεργασία (ή επιστρέφει σφάλμα) όσο ο σηματοφορέας είναι κατειλημμένος από άλλη διεργασία.

- `sem_post()` -- Αύξηση της τιμής του σηματοφορέα.

POSIX (5/6)

- Παρακάτω παρουσιάζονται αναλυτικά οι συναρτήσεις και τα αντίστοιχα ορίσματα των σημαφόρων POSIX.

```
• int sem_init(sem_t *sem, int pshared,
unsigned int value);
```

```
• int sem_destroy(sem_t *sem);
```

POSIX (6/6)

- `sem_t *sem_open(const char *name, int oflag, ...);`
  `int sem_close(sem_t *sem);`
  `int sem_unlink(const char *name);`

- `int sem_wait(sem_t *sem);`

- `int sem_post(sem_t *sem);`

Η συνάρτηση stat() (1/2)

- Υπάρχουν δύο χρήσιμες συναρτήσεις που δίνουν πολλές πληροφορίες για την τρέχουσα κατάσταση ενός αρχείου, στη μορφή μιας σύνθετης δομής. Μπορούμε, για παράδειγμα, να μάθουμε πόσο μεγάλο είναι ένα αρχείο (`st_size`) πότε δημιουργήθηκε (`st_ctime`). Οι συναρτήσεις ορίζονται στη βιβλιοθήκη `sys/stat.h`.

```
int stat(char *path, struct stat *buf),
int fstat(int fd, struct stat *buf)
```

Η συνάρτηση stat() (2/2)

- Η `stat()` παρέχει πληροφορίες για το αρχείο που το όνομά του δίνεται από το path. Δεν απαιτούνται δικαιώματα ανάγνωσης, εγγραφής ή εκτέλεσης στο αρχείο αλλά όλοι οι κατάλογοι στη διαδρομή θα πρέπει να επιτρέπουν ανάγνωση.

- Η `fstat()` θα πάρει τις ίδιες πληροφορίες με βάση τον περιγραφέα αρχείου που επιστρέφεται από την κλήση `open`.

- Οι συναρτήσεις `stat()` και `fstat()` επιστρέφουν 0 στην επιτυχία, `-1` στην αποτυχία και θέτουν το `errno` στον κατάλληλο κωδικό σφάλματος.

Η συνάρτηση lseek() (1/3)

- Συσχετισμένος με κάθε ανοικτό αρχείο είναι ένας δείκτης (file pointer) ο οποίος περιέχει τον αύξοντα αριθμό του byte του αρχείου που θα διαβαστεί (θα γραφτεί) με την επόμενη κλήση `read()`. Ο δείκτης αυτός ξεκινά από την τιμή `0` μόλις ανοίξουμε ένα αρχείο και αυξάνεται αυτόματα μετά από κάθε κλήση `read()`.

- Με την κλήση `lseek()` μας δίνεται η δυνατότητα να δώσουμε στο δείκτη αυτό μια αυθαίρετη τιμή, αναγκάζοντας έτσι τις επόμενες εντολές (`read()`, `(write())`, να διαβάσουν, (να γράψουν), δεδομένα στο επιθυμητό σημείο του αρχείου.

Η συνάρτηση lseek() (2/3)

- Τονίζουμε ότι η κλήση `lseek()` δεν προκαλεί μεταφορά δεδομένων από ή προς το αρχείο.

- Η σύνταξη της κλήσης `lseek()` είναι:

```
#include  <sys/types.h>
#include  <unistd.h>

off_t lseek(int fd, off_t offset, int
whence);
```

Η συνάρτηση lseek() (3/3)

- Η παραπάνω κλήση `lseek()` μετακινεί το δείκτη του αρχείου κατά offset bytes σε σχέση με τη θέση αναφοράς που καθορίζει η τρίτη παράμετρος `whence`. Η παράμετρος `whence` είναι ένας ακέραιος αριθμός που παίρνει τιμές:

- SEEK_SET (=0) Το offset λαμβάνεται σχετικά με την αρχή του αρχείου.
- SEEK_CUR (=1) Το offset λαμβάνεται σχετικά με την τρέχουσα θέση στο αρχείο.
- SEEK_END (=2) Το offset λαμβάνεται σχετικά με το τέλος του αρχείου.

Η συνάρτηση unlink() (1/2)

- Η κλήση `unlink()`, διαγράφει το αρχείο `path` από το σύστημα των αρχείων. Τονίζουμε ότι μόνο το όνομα του αρχείου σβήνεται από τον κατάλληλο κατάλογο. Τα περιεχόμενα του αρχείου στο δίσκο διατηρούνται εφ' όσον υπάρχουν και άλλοι σύνδεσμοι με το αρχείο αυτό και καταστρέφονται μόνον αφού "σπάσουν" όλοι οι σύνδεσμοι που δημιουργήθηκαν με κλήσεις `link()`, εκτελώντας τον αντίστοιχο αριθμό κλήσεων `unlink()`. Στην απλή περίπτωση των αρχείων με τα οποία δεν έχουν δημιουργηθεί άλλοι σύνδεσμοι, η κλήση `unlink()` είναι ισοδύναμη με τις πιο συμβατικές κλήσεις `erase` ή `delete` άλλων ΛΣ.

Η συνάρτηση unlink() (2/2)

`int unlink(cons char *path)`

- Διαγράφει το στοιχείο καταλόγου (αρχείο) με όνομα διαδρομής `path`.

➢ Η `unlink()` επιστρέφει 0 στην επιτυχία, -1 στην αποτυχία και θέτουν το `errno` στον κατάλληλο κωδικό σφάλματος. Οι κωδικοί σφάλματος βρίσκονται στο εγχειρίδιο της πρότυπης βιβλιοθήκης της C (`sys/stat.h`).

Η συνάρτηση basename()

- Η συνάρτηση `basename()` επιστρέφει το όνομα αρχείου από μια διαδρομή. (δηλαδή αν δοθεί το `/usr/local/etc/passwd` θα επιστρέψει μόνο το `passwd`).

- Σύνταξη:

  basename(*path*,*suffix*)

Η συνάρτηση strcat()

- Το πρωτότυπό της βρίσκεται στο <string.h>:

```
char *strcat(char *str1, char *str2)
```

- Η συνάρτηση αυτή συνδέει κατά σειρά τα str1 και str2, τερματίζοντας το str1 με ένα κενό. Το αποτέλεσμα καταχωρείται στο str1. Το str2 παραμένει ανέπαφο από τη διαδικασία.

- Η συνάρτηση strcat() επιστρέφει str1. Πρέπει να αποφύγουμε την περίπτωση που το άθροισμα των στοιχείων των δύο πινάκων να υπερβαίνει τον αριθμό των θέσεων του str1.

Τέλος Ενότητας

Πανεπιστήμιο Δυτικής Μακεδονίας