

10/25/00

Midterm #2

(Wed) November 3rd -

(Thurs) " 4th -

Mon

~~Mon~~ Wed

Tues/Thurs

How to do Destroy Lock?

Difference with Proj 2: Every thread that wants to use a lock MUST do a CreateLock

Have every thread do a DestroyLock, also.

You just need a counter - when

the count is 0 - delete it.

Final Tests for Parts 1 & 2

```
int main() {  
    Exec(matmut);  
    Exec(mat);  
    return 0;  
}
```

Diagram: A curved arrow points from the string "../test/matmut", 15 to the Exec(~~matmut~~) line.

To Fork twice you can't share the arrays

```
int A[20][20]
```

```
" B "
```

```
" C "
```

```
" D "
```

```
" E "
```

```
" F "
```

```
void matmult1() {
```

```
// use A, B, C arrays
```

} original
matmult
code

```
}
```

```
void matmult2() {
```

```
// use D, E, F arrays
```

```
}
```

```
int main() {
```

```
Fork (matmult1);
```

```
Fork (matmult2);
```

```
}
```

About swap file maintenance

- On an Exit for the last thread in a process - you must clear any entries in the swap file for the process.

Protection Systems

The O.S. has the responsibility for the protection of computer resources

General Characteristics

- An OS has resources (objects)
- These resources are managed by

the OS for user programs

- Each resource has a unique identifier
- Each resource has a set of valid operations that can be performed on it

Protection systems have 3
main components

- object/resource to be protected
- set of ~~op~~ allowed operations
- processes (user programs)

Domain Matrix Implementation

A 2D array

Domain: A set of pairs - object/rights
authorized operations

Rules: A user program can only be in one domain at a time

A user program can change their domain

A "typical" domain is the user account.

Each object/rights pairs specifies:

- the object being protected
- the subset of allowable operations the domain can perform

In our matrix:

- A row represents one domain
- A column represents one object



A cell contains the rights for a single domain to a single object

This is a kernel data structure

An example: aludra

averages / $\left(\begin{array}{l} \# \text{ users: } 10,000 \\ \# \text{ files: } 1,000 \end{array} \right.$

Protection Matrix:

(no group/world domain)

10,000 rows.

* 10,000,000 columns -

100,000,000,000 cells

Each domain (1 row) has 10,000,000 elements

but only 1000 entries are used

$$\frac{1000}{10,000,000}$$

\Rightarrow 99.9% of matrix is empty

Better approach: Don't store empty cells - only store rights that exist



No entry means no access

That gives us 2 choices:

- 1) By row (domain) \Rightarrow Protection Domain
- 2) By column (object) \Rightarrow Access Control List

Protection Domain: For each domain, we have a set of pairs (object, rights)

ACL: For each object we have a set of pairs (domain, rights)

ex: Protection Domain

hash table

hash table {
 ~~do~~ user1: (file1, numeric ID), (file2, r),
 user2:
 user3:

ex: ACL

file1: (user1, rw), (user2, x),

file2:

file3:

Which method to choose for an OS?

Protection Domain:

- + Users can see all files they can access
- + Only one active domain, at a time, for a process, we can store a reference to that domain's protection in the process

ACL:

- + Can show all domains which can access a single object

Protection data is privileged (kernel access)

- no direct user access

Issue: At any instant, the protection data determines what a domain can do

⇓

mechanism

It does NOT control what

a domain are AUTHORIZED to do
policy

Protection data may not match the protection Policy

Protection systems only validate against the protection data, NOT the protection policy

Situation

Key ~~Question~~:

- We build a brand new computer system - no user has logged in, yet
- We trust the administrators who are building the system

Protection data matches the protection policy

Question: Can we guarantee that the system will allow be in a state where the policy & data are the same?