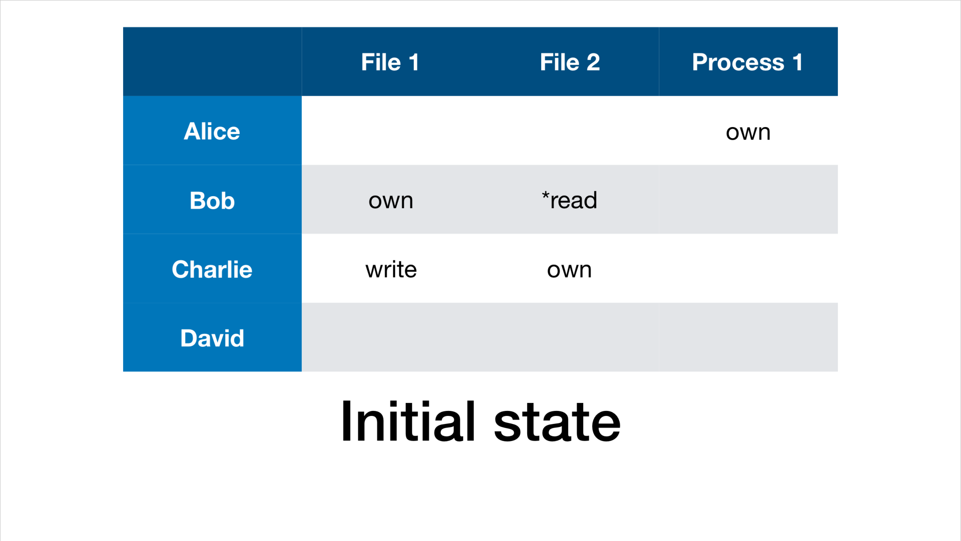
Homework discretionary access control

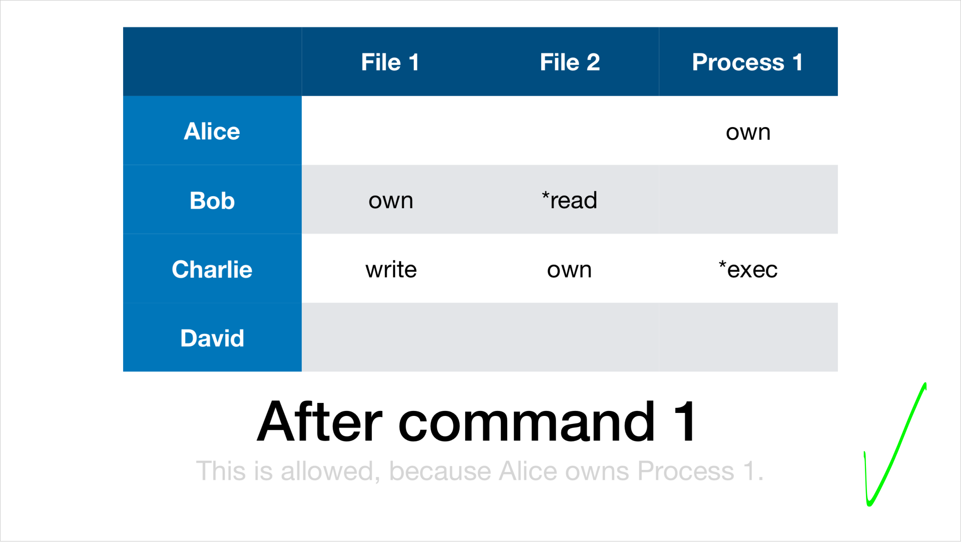
Tuesday, 13 September 2022 10:39

Exercise discretion...



	File 1	File 2	Process 1
Alice			own
Bob	own	*read	
Charlie	write	own	
David			

Initial state



	File 1	File 2	Process 1
Alice			own
Bob	own	*read	
Charlie	write	own	*exec
David			

After command 1

This is allowed, because Alice owns Process 1.

✓

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 2

✓

This is allowed, since Charlie has *exec permission.
Adding exec to Bob for Process 1.

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 3

✓

This will not be allowed, since Bob can only transfer the read right. He cannot confer the *read right, since he does not own File 2. (Note: *read only allows to transfer, not to confer!)

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 4

This command fails; Alice cannot create File 1, since it already exists. ✓

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 5

This command fails, since Alice does not own File 1. (Her creation of it failed, so she is not the owner.)

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 6

This command fails, since Alice does not own File 1. (Her creation of it failed, so she is not the owner.)

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 7

This command fails. Even though Charlie conferred the exec right on Process 1 to Bob, he cannot revoke it, since he does not own Process 1.

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	exec
Charlie	write	own	*exec
David			

After command 8

This command fails. Alice does not have the *read right on File 2, so she cannot transfer this right to David.

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 9

This command succeeds; since Process 2 does not exist yet, Bob is allowed to create it. This also assigns the own right on Process 2 to Bob.

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 10

This command fails. Bob cannot transfer the exec right on Process 2 to Alice, even though he owns the Process. Instead, he should have conferred it (to give Alice that permission).

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 11

This command fails. Bob cannot transfer the exec right on Process 2 to Charlie, even though he owns the Process. Instead, he should have conferred it (to give Charlie that permission).

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 12

This command fails, as Charlie does not have the own right on Process 1, so he cannot revoke any rights on Process 1 (not even his own rights).

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David			exec	

After command 13

This command succeeds. Charlie has the *exec right, and can hence transfer the exec right to David. Note that this introduces a leak, since David has now had a right he did not have at the start for at least one instant.

	File 1	File 2	Process 1	Process 2

Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 14

This command succeeds; Alice has the own right on Process 1, so she can revoke David’s permissions on that Process.

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 15

This command fails; David does not have the read right on File 2, so it cannot be removed.

	File 1	File 2	Process 1	Process 2

Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 16

This command fails; to transfer the write right on File 1, Alice should have the *write right on File 1, which she does not have.

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 17

This command fails; Alice cannot transfer the exec right on Process 2 to Dave, since she does not have the *exec right on that Process.

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 18

This command fails, because David does not have the exec right on Process 2.

	File 1	File 2	Process 1	Process 2
Alice			own	
Bob	own	*read	exec	own
Charlie	write	own	*exec	
David				

After command 19

This command fails, because David does not have the write right on File 1.

	File 1	File 2	Process 1
Alice			own
Bob	own	*read	
Charlie	write	own	
David			

After command 20

This command fails, because of two reasons: Alice does not have the exec right on Process 2, and Bob does not have the own right on Process 2.

Leaks

In this sequence of commands, access privileges are leaked. Note that David obtained the exec right after command 13. Even though that privilege was revoked immediately afterwards in command 14, the fact that he held that privilege for an instant suffices for a leak to have occurred.