

Security Simulator pseudocode
Jake Petroules
CS320 Operating Systems
Spring 2012

Simulates basic security management features of an operating system by means of a terminal interface.

The application begins by prompting for login:

```
IF authentication-is-successful:
    Enter command loop
ELSE
    Display error message and exit
```

```
WHILE in command loop:
    Display a prompt for a command
    Send the command to the interpreter
    IF command is valid:
        Execute the command
    ELSE
        Display an error message
    END IF
END WHILE
```

The command interpreter has a multitude of commands that may be interpreted:

```
Split interpreted command by the space character
Read the first value as the command name
Check that the number of arguments is valid for the given command
```

Run a custom method depending on the name of the command to execute:

```
BEGIN command addgroup:
    IF user is not the root user:
        Permission denied and exit
    END IF

    Check argument count for validity

    Create the group with the given name if it does not exist already
END

BEGIN command adduser:
    IF the user is not the root user:
        Permission denied and exit
    END IF

    Check argument count for validity

    Create the user with the given name if they do not exist already
END

BEGIN command chmod:
    Check argument count for validity

    Check that the resource with the given name exists

    IF user is not the root user AND user does not own the resource being
modified:
        Permission denied and exit
    END IF

    Set the new permissions on the resource
END
```

```
BEGIN command chown:
    IF the user is not the root user:
        Permission denied and exit
    END IF

    Check argument count for validity

    Check that the resource with the given name exists

    Check that the new owner user exists

    Check that the new owner group exists

    If all checks passed, set the new owner user and group if specified
END
```

```
BEGIN command whoami:
    Displays the name of the user who is currently logged in
END
```

```
BEGIN command lsgroups:
    FOR EACH group in the database
        Print the name of the group
    END FOR
END
```

```
BEGIN command ls:
    FOR EACH resource in the database
        IF the -l option was specified:
            Print the resource permissions, owner, group, and name
        ELSE
            Print the resource name
        END FOR
    END FOR
END
```

```
BEGIN command lsusers:
    FOR EACH user in the database
        IF the -l option was specified:
            Print the user name, group list, and date of last password
change
        END IF
    END FOR
END
```

```

BEGIN command modgroup:
    IF user is not the root user:
        Permission denied and exit
    END IF

    Check argument count for validity

    Check that the group with the given name exists

    Check that the user with the given name exists

    IF specified action = add:
        Add the user to the group
    ELSE IF specified action = remove:
        Remove the user from the group
    END IF
END

BEGIN command passwd:
    IF a username was specified:
        IF user is not the root user:
            Permission denied and exit
        ELSE
            Prompt for new password
            Change user's password to new password
        END IF
    ELSE
        Prompt for current password
        IF current password = given password:
            Prompt for new password
            Set new password
        ELSE
            Display error message
        END IF
    END IF
END

```

```

BEGIN command rm:
    Check argument count for validity

    Check that the resource with the given name exists

    IF the current user has write permission to the resource:
        Delete the resource
    ELSE
        Permission denied and exit
    END IF
END

BEGIN command rmgroup:
    IF user is not the root user:
        Permission denied and exit
    END IF

    Check argument count for validity

    Check that the group with the given name exists

    IF name of group to remove = root OR staff
        Display error message explaining that the root and staff groups
cannot be removed
    END IF

    Remove all users from the group and delete the group
END

BEGIN command rmuser:
    IF user is not the root user:
        Permission denied and exit
    END IF

    Check argument count for validity

    IF name of user to remove = root
        Display error message explaining that the root user cannot be
removed
    END IF

    Remove the user
END

```

BEGIN command touch:

Check argument count for validity

IF no resource with the given name exists

Create the resource with the given name

Set the owner of the resource to the current user

IF the current user is a member of one or more groups

Set the group of the resource to the current user's

first/primary group

ELSE

Set the group of the resource to staff

END IF

END IF

END