Section A

CODE:

```
#!/bin/bash
# Define the command version
command_version="v0.1.0"
# Function to display command manual
display_manual() {
  echo "INTENSCTL(1)                    User Commands                    INTENSCTL(1)"
  echo ""
  echo "NAME"
  echo "       internsctl - Custom command for XYZ operations"
  echo ""
  echo "SYNOPSIS"
  echo "       internsctl [OPTIONS]"
  echo ""
  echo "DESCRIPTION"
  echo "       This command performs XYZ operations."
  echo ""
  # Add more sections and descriptions as needed for your command
}

# Function to display command help
display_help() {
  echo "Usage: internsctl [OPTIONS]"
  echo ""
  echo "Options:"
  echo "  --help         Display this help message"
  echo "  --version      Show command version"
  echo ""
  # Add more help information as needed for your command
}

# Parse command-line options
case "$1" in
    --help)
        display_help
        ;;
    --version)
        echo "internsctl $command_version"
        ;;
    *)
        display_manual
        ;;
esac
```

Section B

CODE:

```
#!/bin/bash

# Define the command version
command_version="v0.1.0"

# Function to display command manual
display_manual() {
    echo "INTENSCTL(1)                    User Commands                    INTENSCTL(1)"
    echo ""
    echo "NAME"
    echo "      internsctl - Custom command for XYZ operations"
    echo ""
    echo "SYNOPSIS"
    echo "        internsctl [OPTIONS]"
    echo ""
    echo "DESCRIPTION"
    echo "        This command performs XYZ operations."
    echo ""
}

# Function to display command help
display_help() {
    echo "Usage: internsctl [OPTIONS]"
    echo ""
    echo "Options:"
    echo "  --help          Display this help message"
    echo "  --version       Show command version"
    echo ""

    echo "Part1"
    echo "  internsctl cpu getinfo          Display CPU information (similar to lscpu)"
    echo "  internsctl memory getinfo       Display memory information (similar to free)"
    echo ""
    echo "Part2"
    echo "  internsctl user create <username>       Create a new user with home directory access"
    echo "  internsctl user list                    List all regular users"
    echo "  internsctl user list --sudo-only        List users with sudo permissions"
    echo ""
    echo "Part3"
    echo "  internsctl file getinfo <file-name>    Get detailed file information"
    echo "  Options:"
    echo "    --size, -s              Print file size"
    echo "    --permissions, -p       Print file permissions"
    echo "    --owner, -o            Print file owner"
    echo "    --last-modified, -m      Print last modified time"
    echo ""
    # Add more help information as needed for your command
}

# Function to handle CPU information retrieval
get_cpu_info() {
    lscpu
}

# Function to handle memory information retrieval
get_memory_info() {
    free
}

# Function to create a new user
create_user() {
    if [ "$#" -ne 1 ]; then
        echo "Usage: internsctl user create <username>"
        exit 1
    fi
    sudo useradd -m "$1"
}

# Function to list all regular users
list_users() {
    cut -d: -f1 /etc/passwd
}

# Function to list users with sudo permissions
```

```
list_sudo_users() {
    grep -Po '^sudo.+:\K.*$' /etc/group | tr ',' '\n'
}

# Function to get file information
get_file_info() {
    if [ "$#" -lt 2 ]; then
        echo "Usage: internsctl file getinfo [options] <file-name>"
        exit 1
    fi

    file_name="${@: -1}"  # Get the last argument as the file name

    # Check if the file exists
    if [ ! -e "$file_name" ]; then
        echo "File not found: $file_name"
        exit 1
    fi

    echo "File: $file_name"

    # Parse options
    while [ "$#" -gt 1 ]; do
        case "$1" in
            --size|-s)
                echo -n "Size(B): "
                stat -c "%s" "$file_name"
                ;;
            --permissions|-p)
                echo -n "Access: "
                stat -c "%A" "$file_name"
                ;;
            --owner|-o)
                echo -n "Owner: "
                stat -c "%U" "$file_name"
                ;;
            --last-modified|-m)
                echo -n "Modify: "
                stat -c "%y" "$file_name"
                ;;
            *)
                echo "Invalid option: $1"
                ;;
        esac
        shift
    done
}

# Parse command-line options
case "$1" in
    cpu)
        shift
        get_cpu_info
        ;;
    memory)
        shift
        get_memory_info
        ;;
    user)
        shift
        case "$1" in
```

```
        create)
            shift
            create_user "$@"
            ;;
        list)
            shift
            if [ "$1" = "--sudo-only" ]; then
                list_sudo_users
            else
                list_users
            fi
            ;;
        *)
            display_help
            ;;
    esac
    ;;
file)
    shift
    if [ "$1" = "getinfo" ]; then
        shift
        get_file_info "$@"
    else
        display_help
    fi
    ;;
--help)
    display_help
    ;;
--version)
    echo "internsctl $command_version"
    ;;
  *)
    display_manual
    ;;
esac
```