

N13 - teme lab0

BAJCSI Elias-Robert

2026-02-17 Tue

3

1 Scrieti un script shell care afiseaza lista utilizatorilor si timpul cat au fost logati (din ultima sesiune de lucru)

```
#!/usr/bin/env sh

usage() {
    echo "Usage: ${0}" >&2
    echo "Prints logged users and their logtime" >&2
}

# Check args
[ "$#" -ne 0 ] && { usage; exit 1; }

for user in $(awk -F: '$3 >= 1000 {print $1}' /etc/passwd); do
    session=$(last -n 1 "$user" | head -n 1)
    # Extract duration (the last field in parentheses)
    duration=$(echo "$session" | awk '{print $NF}')
    printf "%-10s %s\n" "$user" "$duration"
done
```

2 Scrieti un script care implementeaza un browser text folosind comenzile wget si links -dump . Utilizatorul are trei optiuni introduce un URL, b pentru back si q pentru quit. Ultimele 10 URL-uri introduse de user sunt stocate intr-un array, de unde utilizatorul poate sa le acceseze folosind functionalitatea back.

```
#!/usr/bin/env sh

usage() {
    echo "Usage: ${0}" >&2
    echo "Simple text-based browser using links and wget" >&2
}

# Check args
[ "$#" -ne 0 ] && { usage; exit 1; }

# Create a unique temporary history file
HISTORY_FILE=$(mktemp) || exit 1
```

```

while ;; do
    echo
    printf "Enter a URL, \b' for back, or \q' to quit: "
    read input

    case "$input" in
        q)
            echo "Exiting browser ..."
            break
            ;;
        b)
            # Check if last page
            COUNT=$(wc -l < "$HISTORY_FILE")
            if [ "$COUNT" -lt 2 ]; then
                echo "No previous page"
                continue
            fi

            # Remove last URL
            head -n $((COUNT - 1)) "$HISTORY_FILE" > "${HISTORY_FILE}.tmp"
            mv "${HISTORY_FILE}.tmp" "$HISTORY_FILE"

            # Get the last URL
            url=$(tail -n 1 "$HISTORY_FILE")
            ;;
        *)
            url="$input"

            # Append URL to history
            echo "$url" >> "$HISTORY_FILE"

            # Keep only last 10 URLs
            tail -n 10 "$HISTORY_FILE" > "${HISTORY_FILE}.tmp"
            mv "${HISTORY_FILE}.tmp" "$HISTORY_FILE"
            ;;
    esac

    # Fetch and display page
    if command -v links >/dev/null 2>&1; then
        links -dump "$url"
    else
        wget -q -O - "$url"
    fi
done

```

3 Scrieti un script care automatizeaza instalarea unui pachet de pe Internet (trebuie dat numele, descarcat, dezarhivat si compilat). Doar instalarea poate fi intrerupta.

```

echo "Attempting to build $PACKAGE_NAME..."

# 0. GNU AutoGen
[ -x "./autogen.sh" ] && {
    echo "Found autogen.sh, running it..."
}

```

```

        ./autogen.sh || echo "autogen.sh failed"
    }

# 1. GNU Autotools
if [ -x "./configure" ]; then
    echo "Found configure script, using autotools..."
    ./configure || echo "Configure failed"
    if make; then
        echo "Autotools build succeeded."
        make install || echo "Installation failed or interrupted."
        exit 0
    fi
fi

# 2. CMake with default generator (Makefiles)
if [ -f "CMakeLists.txt" ]; then
    echo "Found CMakeLists.txt, using CMake with Makefiles..."
    mkdir -p build
    cd build || exit 1
    cmake .. || echo "CMake configuration failed"
    if make; then
        echo "CMake build succeeded."
        make install || echo "Installation failed or interrupted."
        exit 0
    fi
    cd ..
fi

# 3. Plain Ninja
if [ -f "build.ninja" ]; then
    echo "Found build.ninja, using Ninja..."
    ninja || echo "Ninja build failed"
    ninja install || echo "Installation failed or interrupted"
    exit 0
fi

# 4. Plain Makefile
if [ -f "Makefile" ] || [ -f "makefile" ]; then
    echo "Found Makefile, using make..."
    if make; then
        echo "Make build succeeded."
        make install || echo "Installation failed or interrupted."
        exit 0
    fi
fi

# 5. Cargo (Rust)
if [ -f "Cargo.toml" ]; then
    echo "Found Cargo.toml, using Cargo..."
    cargo build --release || echo "Cargo build failed"
    cargo install --path . || echo "Cargo install failed or interrupted"
    exit 0
fi

echo "Error: No recognized build method succeeded."
exit 1

```

4 Scrieti un program C care implementeaza comanda kill.

```
#include <stdio.h>
#include <signal.h>
#include <stdlib.h>
#include <unistd.h>

void term_handler(int);

int main() {
    signal(SIGTERM, term_handler);
    signal(SIGINT, term_handler);

    printf("Haaaiiii! I'm currently going in an infinite loop, until you're going
    to send a SIGTERM, SIGINT or SIGKILL signal.\n");
    fflush(stdout);

    while(1)
        sleep(1);
}

void term_handler(int _) {
    const char msg[] = "Bye :3\n";
    write(STDOUT_FILENO, msg, sizeof(msg) - 1);
    _exit(0);
}
```