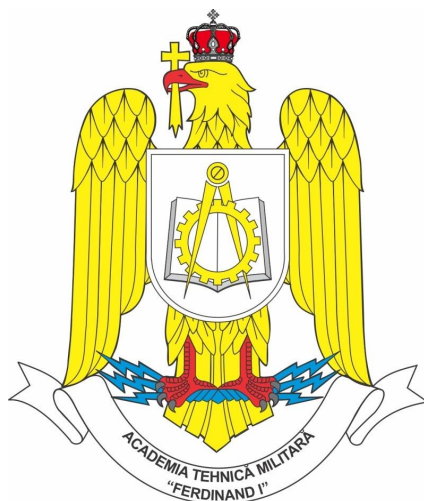


R O M Â N I A
MINISTERUL APĂRĂRII NAȚIONALE
Academia Tehnică Militară "Ferdinand I"



Process energy consumption monitoring

Student Sg. David Ioana Cătălina, gr. C-113D

Student Sg. Lăscuțeanu Ștefan, gr. C-113D

- BUCUREȘTI -

2023

CAPITOLUL 1

INTRODUCERE ȘI CUPRINS

1.1. Scopul documentului

Scopul documentului cu cerințe extinse este de a oferi o descriere amănunțită a programului implementat, respectiv a obiectivelor acestuia. De asemenea, acest document prezintă modul în care acesta se utilizează și diferențele dintre stadiul de proiectare și finalizare.

1.2. Structura documentului

Prezentul document se împarte în 4 capitole, astfel:

- Primul capitol, cel curent, este o scurtă prezentare a obiectivelor pe care acest document și le propune să le realizeze.
- Al doilea capitol își propune să ilustreze o descriere generală a proiectului și o prezentare a platformelor folosite în dezvoltarea aplicației.
- Al treilea capitol prezintă cerințele funcționale și nefuncționale ale sistemului.
- Al patrulea capitol, și ultimul, aduce concluzia și încheierea documentului.

CAPITOLUL 2

DESCRIEREA GENERALĂ A PROIECTULUI

2.1. Descrierea generală a aplicației

Scopul proiectului îl reprezintă o aplicație de monitorizare a consumului energetic al proceselor a câtorva resurse hardware principale: CPU, Memorie, GPU, Baterie. Acest fapt se va realiza prin formule de calcul euristice ce depind de parametrii fizici ai bateriei și resursele ocupate de procese în CPU, GPU și Memorie. Datele de intrare necesare calculelor vor fi obținute din sistemul de fișiere (/proc), comenzi de bash (ex: top, nvidia-smi, turbostat, upower) și funcții POSIX (time(), getsrusage(), etc.).

Deși în stadiu de proiectare, se presupunea existența unei interfețe grafice, aceasta nu a mai fost implementată. În schimb, am adăugat modulul GPU (neprevăzut în stadiul incipient deoarece informațiile despre acesta nu puteau fi obținute într-o mașină virtuală, astfel am hotărât de-a lungul

implementării să configurăm un dual-boot cu Linux) pentru o analiza mai concretă a datelor de interes.

Din punct de vedere arhitectural, aplicația este formată din entitățile operaționale de extragere a datelor și cele computaționale.

2.2. Descrierea platformei de dezvoltare

Programele utilizate pentru dezvoltarea software-ului sunt Visual Studio Code. Totodată, este necesar un sistem de control al versiunii pentru o administrare și documentare bună a modificărilor aduse proiectului, precum și pentru facilitarea colaborării între dezvoltatori. În acest sens a fost utilizată platforma Git împreună cu GitHub.

CAPITOLUL 3 **DETALIEREA CERINTELOR SPECIFICE**

3.1. Cerințe funcționale

Principalele elemente de interfață cu care utilizatorul poate interacționa sunt: terminalul pus la dispoziție de Linux.

3.1.1. Obținerea listei de procese în timp real

Acest procedeu va fi realizat prin intermediul comenzii „top”, al cărei output va fi redirecționat către celelalte componente.

3.1.2. Obținerea datelor de intrare

Toate informațiile necesare despre CPU, memorie și baterie se regăsesc în /proc/cpuinfo/, /proc/meminfo/, respectiv în /sys/class/power_supply/. De asemenea, a fost necesară instalarea pachetelor: turbostat pentru obținerea wattajului nominal al componentelor, nvidia-smi pentru monitorizarea detaliată a GPU, dmi-decode pentru memorie. Întrucât comanda turbostat face o analiză îndelungată pentru a obține informații la momente cheie de timp, le-am rulat în prealabil și am salvat în fișierele cpu/mem/gpu_pck_wat.txt rezultatele obținute, urmând ca mai apoi să se facă o medie aritmetică în cod pentru fiecare în parte.

```

PkgWatt --interval 5 > cpu_pck_wat
^Cioana@ioana-Lenovo-Legion-Y530-15ICH:~$ sudo turbostat --Summary --quiet --sho
w PkgWatt --interval 5 > cpu_pck_wat.txt
^Cioana@ioana-Lenovo-Legion-Y530-15ICH:~$ cat cpu_pck_wat.txt
PkgWatt
11.65
11.42
11.47
11.36
11.38
11.37
11.37
11.32
11.32

```

```

ioana@ioana-Lenovo-Legion-Y530-15ICH:~$ sudo turbostat --Summary --quiet --show
RAMWatt --interval 5 > mem_pck_wat.txt
ioana@ioana-Lenovo-Legion-Y530-15ICH:~$ cat mem_*
RAMWatt
0.47
0.54
0.55
0.70
0.51
0.43
0.43
0.49
0.45
0.48
0.35
0.35
0.41
0.51

```

```

0.02
ioana@ioana-Lenovo-Legion-Y530-15ICH:~$ sudo turbostat --Summary --quiet --show
GFXWatt --interval 5 > gpu_pck_wat.txt
ioana@ioana-Lenovo-Legion-Y530-15ICH:~$ cat gpu_*
GFXWatt
0.11
0.22
0.21
0.15
0.09
0.15
0.04
0.03

```

3.1.3. Work-flow-ul programului

Programul pornește în main.cpp care presupune un meniu pentru selectarea acțiunii dorite: show all/ doar un proces specific / exit. Selectarea uneia dintre cele două variante presupune inițializarea componentelor individuale, inițializarea listei de procese, efectuarea calculelor și afișarea rezultatelor.

Inițializarea componentelor presupune obținerea informațiilor specifice pentru fiecare în parte, stocate apoi la nivel de entitate (fiecare componentă este de tip Singleton). Pentru informațiile utilizate de procese, acestea se inițializează la nivel individual, la cererea clasei hwman, pe când pentru informațiile relevante pentru baterie acestea se inițializează doar la nivel individual și sunt apelate mai apoi în main.

Inițializarea proceselor presupune obținerea listei de procese (doar 50 pentru viteza programului cu filtru de grad de utilizare) și calcularea, la nivel de proces, a ratei de utilizare per fiecare componentă în parte (CPU, MEM, GPU).

Clasa Procman se ocupă cu pornirea procesului de populare a listei de procese și a datelor specifice la nivel de proces: datele privind consumul per resursă.

Efectuarea calculelor din main presupune calcularea efectivă a consumului fiecărui proces relativ la baterie, utilizând datele obținute în urma apelării funcțiilor de inițiere mai sus prezentate.

3.1.4. Formule matematice

Formulele matematice principale pe care aplicația le va folosi pentru obținerea calcului energiei sunt:

- $\text{Process CPU Power} = \text{Global CPU Power} * \text{Process CPU Usage}$,
unde $\text{Process CPU Usage} = \text{Process CPU Time} / \text{Global CPU Time}$.
- $\text{CPU_usage of battery} = \text{average_cpu_power} * \text{BATTERY_power}/100$
- $\text{MEM_usage of battery} = \text{average_mem_power} * \text{BATTERY_power}/100$
- $\text{GPU_usage of battery} = \text{average_gpu_power} * \text{BATTERY_power}/100$
- $\text{Process_usage of battery} = \text{process_usage_of_components} * \text{battery_power}/100$;
- unde $\text{process_usage_of_components} = \text{process_usage_of_cpu} * \text{CPU_usage_of_battery} / 100 + \text{process_usage_of_gpu} * \text{GPU_usage_of_battery} / 100 + \text{process_usage_of_mem} * \text{MEM_usage_of_battery} / 100$

Evident că, în lipsa unor senzori integrați sau a unor dispozitive externe de măsurare a energiei reale consumate, calculele sunt aproximative și orientative, scopul aplicației fiind acela de a contura o imagine de ansamblu asupra ratei de consum energetic a unui proces comparativ cu restul proceselor.

3.3. Cerințe nefuncționale

- *Cerințe de implementare*: utilizarea limbajului C/C++
- *Cerințe de interfață*: aplicația va putea rula pe sistemele de operare Linux. (nu este suportată rularea pe mașini virtuale)
- *Cerințe de securitate*: aplicația nu va modifica sub nicio formă procesele monitorizate sau sistemul de fișiere.
- *Cerințe de utilizare*: sunt necesare cunoștințe minime ale noțiunilor ce țin de componente hardware și de operabilitatea sistemelor de operare.
- *Cerințe exacte și de precizie*: calculele matematice să fie cât mai precise.
- *Cerințe de audit și control*: mecanism de control al comportamentelor necorespunzătoare și mecanisme de împiedicare ale acestora.

CAPITOLUL 4 **CONCLUZII**

Acest livrabil prezintă succint modul de implementare al aplicației, workflow-ul acesteia și evoluția proiectului.

Concluziile trase în urma desfășurării proiectului: domeniul de energy consumption la nivel de proces este extraordinar de limitat din punct de vedere al sistemelor de operare. În momentul de față sistemele de operare nu prezintă facilități extinse și detaliate pentru un bun tracking al modului în care este disipată energia. În lipsa unor senzori dedicați în acest sens, calculele sunt euristice, limitate la informațiile furnizate de producător pentru file-system, dependente de module de urmărire în timp a câtorva componente majore și influențate de absența informațiilor relevante pentru restul componentelor și interoperabilitatea acestora.

De asemenea, în urma rulării programului, conform așteptărilor noastre, consumul energetic al unui proces individual este extraordinar de mic.