OS Dashboard - Documentação

Índice

Visão Geral

O **OS Dashboard** é um sistema de monitoramento em tempo real para sistemas operacionais Linux que coleta e exibe informações detalhadas sobre:

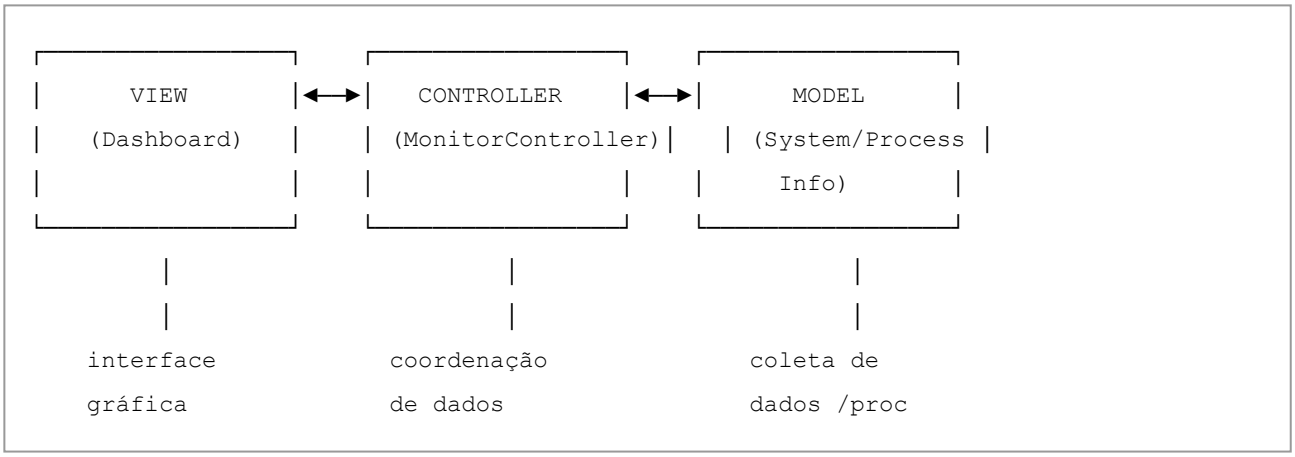
- **CPU**: Uso em tempo real, gráficos históricos
- **Memória**: RAM, cache, buffers, swap
- **Processos**: Lista completa, detalhes, consumo de recursos
- **Threads**: Threads ativas do sistema

Tecnologias Utilizadas

- **Python 3.11**
- **Tkinter**
- **Matplotlib**

Arquitetura do Sistema

Padrão MVC (Model-View-Controller)



Fluxo de Dados

1. **Model Layer**: Coleta dados do sistema via `/proc`
2. **Controller Layer**: Processa e coordena a coleta
3. **View Layer**: Exibe informações na interface gráfica

Estrutura de Arquivos

```
operating_system_dashboard/
├── main.py                     # ponto de entrada da aplicação
├── controller/
│   └── monitor_controller.py   # controlador principal
├── model/
│   ├── system_info.py          # coleta dados de CPU/memória e disco
│   ├── process_info.py         # coleta dados de processos e threads
│   └── file_info.py            # coleta dados de arquivos e diretórios
├── view/
│   ├── dashboard.py            # interface principal
│   └── utils.py               # utilitários de formatação
```

Descrição dos Arquivos

Arquivo	Responsabilidade
`main.py`	início da aplicação e tratamento de erros
`monitor_controller.py`	coordena coleta de dados em thread separada
`system_info.py`	coleta dados de CPU, memória, disco e recursos de processos via `/proc`
`process_info.py`	coleta dados de processos e threads via `/proc`
`file_info.py`	coleta informações detalhadas de arquivos e diretórios
`dashboard.py`	interface gráfica com Tkinter e Matplotlib
`utils.py`	funções auxiliares para formatação

Telas e Funcionalidades

Aba GLOBAL

Elementos Visuais:

- **Card "Uso da CPU"**: Percentual atual de uso da CPU
- **Card "Tempo Ocioso"**: Percentual de tempo que a CPU ficou ociosa
- **Card "Processos"**: Contagem total de processos no sistema
- **Card "Threads"**: Contagem total de threads no sistema
- **Gráfico de CPU**: Histórico de uso da CPU em tempo real

Fontes de Dados:

- `/proc/stat`: Estatísticas da CPU (user, system, idle, iowait, etc.)
- `/proc/*/status`: Contagem de processos e threads

Cálculos:

```
# Percentual de CPU
cpu_usage = (delta_total - delta_idle) / delta_total * 100

# Tempo ocioso
idle_percentage = 100 - cpu_usage
```

Aba PROCESSOS

Cards de Métricas

- **"TOTAL DE PROCESSOS"**: Número de diretórios numéricos em `/proc`
- **"TOTAL DE THREADS"**: Soma do campo `Threads` de todos os processos

Sub-aba "PROCESSOS ATIVOS"

Tabela com colunas:

- **PID**: ID do processo (nome do diretório em `/proc`)
- **USUÁRIO**: Nome do usuário (mapeado de UID via `/etc/passwd`)
- **PROCESSO**: Nome do executável (campo `Name` de `/proc/PID/status`)
- **STATUS**: Estado do processo (campo `State` de `/proc/PID/status`)
- **MEMÓRIA**: Uso de RAM (campo `VmRSS` de `/proc/PID/status`)
- **THREADS**: Número de threads (campo `Threads` de `/proc/PID/status`)

Ordenação: Por uso de memória (decrescente)

Sub-aba "THREADS ATIVAS"

Tabela com colunas:

- **TID**: Thread ID (diretórios em `/proc/PID/task`)
- **PID**: Process ID pai
- **USUÁRIO**: Usuário proprietário do processo
- **PROCESSO**: Nome do processo pai
- **STATUS**: Estado da thread (de `/proc/PID/task/TID/status`)

Sub-aba "DETALHES"

Informações Exibidas:

- **Informações Básicas**: Nome, Estado, PID, PPID, UID, GID, Threads
- **Informações de Memória**: VmPeak, VmSize, VmRSS, VmData, etc.
- **Uso de Páginas**: Total, Código, Heap, Stack (de `/proc/PID/smaps`)
- **Linha de Comando**: Comando completo (de `/proc/PID/cmdline`)

Aba MEMÓRIA

Painel de Métricas

MEMÓRIA FÍSICA:

- **Total**: `MemTotal` de `/proc/meminfo`
- **Em Uso**: `MemTotal - MemFree - Buffers - Cached`
- **Livre**: `MemFree` de `/proc/meminfo`
- **% Uso**: `(memória_usada / total) * 100`

CACHE/BUFFER:

- **Cache**: `Cached` de `/proc/meminfo`
- **Buffers**: `Buffers` de `/proc/meminfo`

SWAP:

- **Swap Total**: `SwapTotal` de `/proc/meminfo`

Detalhes Completos (Botão "Exibir Mais")

Exibe todas as métricas de `/proc/meminfo`.

Gráfico em Tempo Real

- Linha: Percentual de uso de memória ao longo do tempo
- Zonas coloridas: Laranja (80-90%), Vermelho (90-100%)

Aba SISTEMA DE ARQUIVOS

Exibe partições, uso de disco e permite navegação por diretórios.

Aba DIRETÓRIOS

Permite explorar a árvore de diretórios, visualizar arquivos, permissões, proprietário, grupo, tamanho e tipo.

Busca de Arquivos

Busca arquivos por padrão de nome em diretórios selecionados, exibindo detalhes como nome, caminho, tamanho e permissões.

Gráfico em Tempo Real

- **Linha**: Percentual de uso de memória ao longo do tempo
- **Zonas coloridas**:
  - Laranja (80-90%): Zona de atenção
  - Vermelho (90-100%): Zona crítica

Fontes de Dados

Sistema de Arquivos `/proc`

O dashboard utiliza exclusivamente o sistema de arquivos `/proc` do Linux:

Arquivo/Diretório	Dados Extraídos
`/proc/stat`	Estatísticas da CPU (user, system, idle)
`/proc/meminfo`	Informações detalhadas de memória
`/proc/[PID]/status`	Status completo de cada processo
`/proc/[PID]/cmdline`	Linha de comando do processo
`/proc/[PID]/smaps`	Mapeamento detalhado de memória
`/proc/[PID]/task/`	Threads de cada processo
`/etc/passwd`	Mapeamento UID → nome de usuário

Atualização de Dados

- **Frequência**: 1 segundo (configurável)
- **Thread Separada**: Coleta não bloqueia interface
- **Cache Inteligente**: UID→username cached para performance

---

Métricas e Cálculos

CPU Usage

```
# Leitura atual e anterior de /proc/stat
delta_total = total_time_now - total_time_before
delta_idle = idle_time_now - idle_time_before
cpu_usage = (delta_total - delta_idle) / delta_total * 100
```

Memory Usage

```
# Memória efetivamente usada
used_memory = MemTotal - MemFree - Buffers - Cached
memory_percentage = (used_memory / MemTotal) * 100
```

Process Memory

```
# VmRSS já está em kB no /proc/PID/status
memory_bytes = VmRSS_kb * 1024
formatted = format_memory_size(memory_bytes)  # Converte para MB/GB
```
