Arquiva Email GUI - Documentação Completa

Objetivos e Funcionalidades

O `arquiva_email_gui.py` é uma ferramenta especializada para organização automática de arquivos, com foco principal em mensagens de email (`.eml`). O programa:

- Organiza automaticamente arquivos em uma estrutura cronológica de pastas (Ano/Ano-Mês)
- Extrai datas de cabeçalhos de emails para arquivos `.eml`
- Utiliza datas de modificação para outros tipos de arquivo
- Sanitiza nomes de arquivos removendo caracteres inválidos e prefixos desnecessários
- Resolve conflitos de nomes automaticamente
- Fornece feedback detalhado sobre o processo de arquivamento

Modo de Usar

1. **Execução do Programa**:
    - Execute o arquivo `arquiva_email_gui.exe` (ou o nome que o executável recebeu).
    - Uma mensagem informativa inicial será exibida, seguida por uma janela de diálogo para seleção da pasta.
2. **Seleção da Pasta**:
    - Selecione a pasta que contém os arquivos a serem organizados
    - O programa usará esta mesma pasta como raiz para criar a estrutura de arquivamento
3. **Processamento**:
    - O programa processará automaticamente todos os arquivos na pasta selecionada
    - Os arquivos serão movidos para subpastas no formato `Ano/Ano-Mês` baseado em suas datas
    - Uma janela de resumo com contagem de arquivos processados e erros, e um contador regressivo para auto-fechamento, será exibida ao final.
4. **Verificação de Resultados**:
    - Após a conclusão, verifique a estrutura de pastas criada
    - Se ocorreram erros, consulte os logs na pasta "ERROS" criada na raiz

Detalhes Técnicos

Estrutura de Arquivamento

- **Formato de Pastas**: `Ano/Ano-Mês` (exemplo: `2023/2023-05`)
- **Localização**: As pastas são criadas dentro da pasta selecionada pelo usuário

Processamento de Datas

Para arquivos `.eml`:

1. Tenta extrair o cabeçalho `Date` do email
2. Prioriza a análise da data usando `email.utils.parsedate_to_datetime`, que é robusta para diversos formatos e fusos horários. Datas com fuso horário são convertidas para o fuso horário local e depois tornadas "naive" (sem informação de fuso).
3. Se a análise inicial falhar, tenta uma lista de formatos de data comuns (`strptime`), após limpar a string de data de informações de fuso horário entre parênteses (ex: `(UTC)`) ou sufixos como `UTC`/`GMT`.
4. Em caso de falha em todas as tentativas de extração e análise da data, utiliza a data e hora atuais como fallback para determinar a pasta de arquivamento.

Para outros arquivos:

- Utiliza a data de modificação do arquivo obtida via `os.path.getmtime()`
- Se houver erro ao obter a data de modificação, um erro é registrado no log e o arquivo não é movido.

Tratamento de Nomes de Arquivo

1. **Sanitização**:
    - Remove o prefixo `"msg "` (comum em emails exportados)
    - Substitui caracteres inválidos (`< > : " / \ | ? *`) por underscores
    - Remove caracteres de controle (ASCII 0-31)
    - Remove espaços em branco no início ou fim do nome.
    - Normaliza números no início do nome para remover zeros à esquerda
    - Se o nome do arquivo ficar vazio após a sanitização, utiliza "arquivo_renomeado" como fallback.
2. **Truncamento**:
    - Garante que o caminho completo do arquivo de destino (pasta + nome do arquivo) não exceda um limite seguro (aproximadamente 249 caracteres, derivado de `MAX_PATH_LENGTH = 259` e `SAFE_FILENAME_MARGIN = 10`).
    - Preserva a extensão original do arquivo
    - O truncamento é feito na parte do nome do arquivo, considerando o comprimento do caminho da pasta de destino e da extensão.
    - Utiliza codificação UTF-8 para calcular os comprimentos e tenta evitar quebrar caracteres multibyte no meio.
3. **Resolução de Conflitos**:
    - Se um arquivo com o mesmo nome já existir no destino: a. Tenta adicionar um sufixo numérico incremental (ex: `_1`, `_2`, ..., até `_100`). b. Se o nome com sufixo numérico ainda for muito longo (excedendo o limite de caminho), ou se o limite de tentativas com contador for atingido: i. Pega o nome do arquivo original (antes de adicionar o contador), anexa um timestamp detalhado (com microssegundos). ii. Trunca este novo nome (com timestamp) se necessário para caber no limite de caminho. iii. Se mesmo após isso houver colisão (extremamente raro), o arquivo não é movido e um erro é logado.
    - O objetivo é sempre encontrar um nome único que respeite os limites de comprimento do caminho.

Sistema de Logging

- **Nível**: Configurado para registrar apenas mensagens de nível `ERROR`.
- **Localização**: Cria pasta "ERROS" na raiz selecionada
- **Formato**: Arquivos de log com timestamp único (`archive_failures_YYYYMMDDHHMMSS.log`)
- **Conteúdo**: Detalhes específicos sobre cada erro ocorrido durante o processamento

Interface Gráfica

- **Tecnologia**: Implementada com `tkinter`
- **Componentes**:
  - Diálogo de seleção de pasta
  - Mensagens informativas (`messagebox`) no início e para confirmação da pasta selecionada.
  - Janela de resumo com auto-fechamento ao final do processamento
  - Contadores de arquivos processados e erros encontrados

Casos de Uso

Organização de Backups de Email

Ideal para arquivos `.eml` exportados de clientes de email, mantendo a cronologia original baseada no cabeçalho "Date".

Arquivamento de Documentos

Organiza documentos baseados em sua data de modificação, facilitando a localização por período.

Consolidação de Backups

Unifica múltiplos backups em uma estrutura consistente, evitando duplicação através da resolução de conflitos.

Migração entre Sistemas de Email

Prepara emails exportados para importação em novo sistema, mantendo a organização cronológica.

Limitações e Considerações

- **Processamento de Subpastas**: O programa processa apenas arquivos no nível raiz da pasta selecionada
- **Movimentação vs. Cópia**: Utiliza `shutil.move()`, que remove o arquivo original da pasta de origem
- **Limites de Sistema de Arquivos**: O programa tenta ativamente evitar caminhos completos que excedam aproximadamente 249 caracteres (no Windows). Nomes de arquivo podem ser truncados para respeitar este limite.
- **Dependência de Formatos de Data**: A precisão da organização depende dos formatos de data nos cabeçalhos dos emails

Requisitos Técnicos

- **Sistema Operacional**: O executável (`.exe`) é projetado para rodar em sistemas Windows.
- **Dependências**: Nenhuma instalação adicional é necessária para executar o arquivo `.exe`, pois todas as dependências (como Python e bibliotecas necessárias) estão empacotadas nele.

Dicas de Uso

- Faça um backup dos arquivos antes de executar o programa pela primeira vez
- Para arquivos `.eml`, verifique se os cabeçalhos de data estão em formatos padrão
- Evite nomes de arquivo extremamente longos para prevenir truncamento excessivo
- Verifique os logs de erro após o processamento para identificar possíveis problemas