

## Trabalho do Grau B | Compactador Paralelo com Processos e Threads

### Objetivo

Implementar, em C (Linux), um programa que utilize processos e threads cooperando entre si para realizar a compactação e descompactação de texto, explorando conceitos de execução paralela, comunicação, sincronização, gerenciamento de recursos, uso correto da memória compartilhada entre threads e na leitura e gravação em disco de forma paralela.

O ponto de partida são os arquivos fornecidos (mini\_zip.c e os textos de entrada). A partir deles, o grupo deverá fazer o programa funcionar em paralelo, dividindo o trabalho entre diferentes processos e threads.

---

### Execução

O programa deve envolver **três processos principais**:

1. **Compactador** – responsável por ler o arquivo de entrada, processar (compactar) e gravar a saída. Deve utilizar **três threads** trabalhando em paralelo:
  - o uma para **leitura** do arquivo,
  - o uma para **compactação** (processamento dos dados),
  - o e uma para **gravação** em disco.
2. **Descompactador** – responsável por ler o arquivo compactado e reconstruir o texto original.
3. **Monitor** – que exibe **informações gerais da execução**, como total de palavras lidas, palavras compactadas, palavras não encontradas no dicionário, tamanho original e compactado, e percentual final de compressão.

Os três processos devem **executar de forma concorrente** e **compartilhar informações** por algum mecanismo de comunicação entre processos.

A escolha das estruturas e estratégias de comunicação fica **livre** para cada grupo.

---

### Requisitos técnicos (obrigatórios)

- Implementar **três processos distintos** executando em paralelo.
  - No processo de compactação, implementar **três threads** que atuem simultaneamente em leitura, processamento e gravação.
  - Utilizar **estruturas adequadas** para comunicação entre processos e sincronização entre threads.
  - O programa deve permitir **executar os dois cenários de teste** (com os dois arquivos fornecidos), exibindo:
    - o o conteúdo original,
    - o o arquivo compactado,
    - o e o arquivo descompactado resultante.
  - A execução deve demonstrar que o processo ocorre de forma paralela e correta.
- 

### Forma de entrega

O grupo pode escolher entre:

- **Entrega online:** enviar os **códigos-fonte** (todos os arquivos usados) no Moodle e um **vídeo de até 20 minutos**.
- **Apresentação presencial:** enviar os **códigos-fonte** (todos os arquivos usados) no Moodle e apresentar o trabalho diretamente ao professor, mostrando o código, a compilação e as execuções.

---

### Apresentação (vídeo ou presencial) — até 20 minutos

Cada grupo deve apresentar:

1. **Apresentação pessoal** – todos os integrantes com nome completo.
2. **Descrição geral do funcionamento** – explicação breve sobre o papel de cada processo e das threads no processo de compactação:
  - Como os **processos trocam informações**.
  - Como as **threads trabalham em conjunto** para compactar.
  - Quais **dificuldades técnicas** foram enfrentadas e como foram resolvidas.
3. **Demonstração prática:**
  - Compilação dos programas.
  - Execução com os dois arquivos de teste, mostrando:
    - o arquivo original,
    - o resultado da compactação,
    - e o resultado da descompactação.

---

### Desafio (obrigatório para nota máxima)

Utilizar um **dicionário maior**, capaz de abranger todas as palavras presentes nos textos de teste. Essa ampliação deve ser **demonstrada explicitamente** na apresentação. Sem esta implementação, a nota máxima será **9 pts**.

---

### Avaliação (pontuação)

Critério	Descrição
<b>4 pts — Código e implementação técnica</b>	Corresponde à correta utilização de processos e threads, execução realmente concorrente, uso adequado de memória compartilhada e comunicação entre processos. O código deve compilar e funcionar nos dois cenários de teste, mostrando coerência técnica e organização.
<b>3 pts — Demonstração e explicação</b>	Clareza na apresentação do funcionamento do programa, explicando como os processos e threads se coordenam, como ocorre a leitura e gravação em paralelo e quais estratégias foram usadas para comunicação e sincronização. Todos os integrantes devem participar e demonstrar domínio sobre o que foi implementado.
<b>2 pts — Organização e boas práticas</b>	Estrutura geral do projeto, legibilidade do código, comentários objetivos, uso correto de funções. A explicação deve refletir entendimento dos conceitos utilizados de paralelismo e compartilhamento de memória.

Critério	Descrição
<b>1 pt — Desafio (nota máxima)</b>	Implementação e demonstração de um dicionário ampliado, capaz de abranger todas as palavras dos textos de teste. Esse recurso deve ser claramente mostrado na apresentação.

---

#### Observações importantes

- O envio dos **códigos-fonte** é obrigatório.
- Todos os integrantes devem **participar da apresentação** (vídeo ou presencial).
- Trabalhos **idênticos ou copiados** terão nota zero para todos os envolvidos.
- Sem o código fonte ou apresentação (em vídeo ou presencial) o trabalho terá nota zero.