

INF 1316

Trabalho 2 - gerência de memória

Érica Regnier - 2211893

Hanna Epelboim Assunção - 2310289

Código:

```
/*  
Érica Regnier - 2211893  
Hanna Epelboim Assunção - 2310289  
*/  
  
#include <stdbool.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#define MAX_FRAMES 1024  
  
typedef struct {  
    unsigned int pgNum;  
    bool referenciada;  
    bool modificada;  
    unsigned int ultAcesso;  
    unsigned int carregada;  
} Frame;  
  
Frame frames[MAX_FRAMES];  
  
unsigned int clockTime = 0;  
unsigned int framesNum = 0;  
unsigned int pf = 0;  
unsigned int pwSujas = 0;  
unsigned int ptrSegundaChance = 0;  
static unsigned int nruCount = 0;  
  
int substituiNRU(unsigned int framesNum) {  
  
    for (unsigned int i = 0; i < framesNum; i++) {  
        if (frames[i].pgNum == -1) {
```

```

        return i;
    }
}

for (unsigned int i = 0; i < framesNum; i++) {
    if (!frames[i].referenciada && !frames[i].modificada) {
        return i;
    }
}
for (unsigned int i = 0; i < framesNum; i++) {
    if (!frames[i].referenciada && frames[i].modificada) {
        return i;
    }
}
for (unsigned int i = 0; i < framesNum; i++) {
    if (frames[i].referenciada && !frames[i].modificada) {
        return i;
    }
}
for (unsigned int i = 0; i < framesNum; i++) {
    if (frames[i].referenciada && frames[i].modificada) {
        return i;
    }
}
// return rand() % framesNum;
}

int substituiLRU(unsigned int framesNum) {
    unsigned int maisVelho = clockTime;
    int substituido = 0;
    for (unsigned int i = 0; i < framesNum; i++) {
        if (frames[i].ultAcesso < maisVelho) {
            maisVelho = frames[i].ultAcesso;
            substituido = i;
        }
    }
    return substituido;
}

int substituiClock(unsigned int framesNum) {

```

```

while (1) {
    if (!frames[ptrSegundaChance].referenciada) {
        int substituido = ptrSegundaChance;
        ptrSegundaChance = (ptrSegundaChance + 1) % framesNum;
        return substituido;
    } else {
        frames[ptrSegundaChance].referenciada = 0;
        ptrSegundaChance = (ptrSegundaChance + 1) % framesNum;
    }
}

int substituiSecondChance(unsigned int framesNum) {
    unsigned int maisVelho = clockTime;
    int substituido = 0;
    while (1) {
        for (unsigned int i = 0; i < framesNum; i++) {
            if (frames[i].carregada < maisVelho) {
                maisVelho = frames[i].carregada;
                substituido = i;
            }
        }
        if (!frames[substituido].referenciada) {
            return substituido;
        } else {
            frames[substituido].referenciada = 0;
            frames[substituido].carregada = clockTime;
            frames[substituido].ultAcesso = clockTime;
        }
    }
}

int substituiOtimo(unsigned int framesNum, unsigned int *referencias,
                  unsigned int totalRefs, unsigned int posAtual) {
    unsigned int maisDistante = 0;
    int substituido = -1;

    for (unsigned int i = 0; i < framesNum; i++) {
        if (frames[i].pgNum == -1) {

```

```

        // printf("%d pgNA\n", frames[i].pgNum);
        return i;
    }
}

for (unsigned int i = 0; i < framesNum; i++) {
    unsigned int pgNumAtual = frames[i].pgNum;
    bool encontrada = false;
    unsigned int distancia = 0;
    for (unsigned int j = posAtual + 1; j < totalRefs; j++) {
        if (referencias[j] == pgNumAtual) {
            encontrada = true;
            distancia = j - posAtual;
            break;
        }
    }
    if (!encontrada) {
        return i;
    }
    if (distancia > maisDistante) {
        maisDistante = distancia;
        substituido = i;
    }
}
return substituido;
}

int buscaPgFrame(unsigned int pgNum, unsigned int framesNum) {
    for (unsigned int i = 0; i < framesNum; i++) {
        if (frames[i].pgNum == pgNum) {
            return i;
        }
    }
    return -1;
}

void inicializa(unsigned int framesNum) {
    for (unsigned int i = 0; i < framesNum; i++) {
        frames[i].pgNum = -1;
        frames[i].referenciada = false;
    }
}

```

```

        frames[i].modificada = false;
        frames[i].ultAcesso = 0;
        frames[i].carregada = 0;
    }
}

void simulador(const char *alg, const char *arqNome, unsigned int
pgTam_kb,
               unsigned int memTam_mb) {
    printf("Executando o simulador...\n");
    printf("Arquivo de entrada: %s\n", arqNome);
    printf("Tamanho da memoria fisica: %dMB\n", memTam_mb);
    printf("Tamanho das páginas: %dKB\n", pgTam_kb);
    printf("Algoritmo de substituição: %s\n", alg);

    FILE *entrada = fopen(arqNome, "r");
    if (!entrada) {
        perror("Erro ao abrir o arquivo de entrada");
        return;
    }

    unsigned int totalRefs = 0;
    unsigned int pgTam = pgTam_kb * 1024;
    unsigned int memTam = memTam_mb * 1024 * 1024;
    unsigned int addr;
    char rw;

    framesNum = memTam / pgTam;
    inicializa(framesNum);
    unsigned int addrCount = 0;

    while (fscanf(entrada, "%x %c", &addr, &rw) != EOF) {
        addrCount++;
    }

    unsigned int *referencias =
        (unsigned int *)malloc(sizeof(unsigned int) * addrCount);
    rewind(entrada);

    unsigned int idx = 0;
    char rwTmp;

```

```
while (fscanf(entrada, "%x %c", &addr, &rwTmp) != EOF) {
    referencias[idx++] = addr / pgTam;
}

// Agora, vamos rodar a simulação
rewind(entrada);

int posAtual = -1;
while (fscanf(entrada, "%x %c", &addr, &rw) != EOF) {
    posAtual++;
    clockTime++;
    unsigned int pgNum = addr / pgTam;
    int frameIndex = buscaPgFrame(pgNum, framesNum);
    if (strcmp(alg, "NRU") == 0) {
        if (nruCount == 4) {
            for (unsigned int i = 0; i < framesNum; i++) {
                frames[i].referenciada = 0;
            }
            nruCount = 0;
        } else {
            nruCount++;
        }
    }
    if (frameIndex == -1) {
        pf++;
        int substituido;

        if (strcmp(alg, "NRU") == 0) {
            substituido = substituiNRU(framesNum);
        } else if (strcmp(alg, "LRU") == 0) {
            substituido = substituiLRU(framesNum);
        } else if (strcmp(alg, "SecondChance") == 0) {
            substituido = substituiSecondChance(framesNum);
        } else if (strcmp(alg, "Clock") == 0) {
            substituido = substituiClock(framesNum);
        } else if (strcmp(alg, "Optimal") == 0) {
            substituido =
                substituiOtimo(framesNum, referencias, addrCount, posAtual);
        } else {
            printf("ERRO: Algoritmo de substituição inválido.\n");
        }
    }
}
```

```

    }

    if (frames[substituido].modificada) { // alteracao do disco
        pwSujas++;
        frames[substituido].modificada = false;
    }

    // substituicao
    frames[substituido].pgNum = pgNum;
    frames[substituido].referenciada = true;
    if (rw == 'W') {
        frames[substituido].modificada = true;
    }
    frames[substituido].ultAcesso = clockTime;
    frames[substituido].carregada = clockTime;

}

else {
    frames[frameIndex].referenciada = true;
    if (rw == 'W') {
        frames[frameIndex].modificada = true;
    }
    frames[frameIndex].ultAcesso = clockTime;
}
}
fclose(entrada);

/*
for(int i = 0; i<framesNum; i++){
    if(frames[i].modificada){
        pwSujas++;
    }
}
*/

printf("Relatório de Simulação:\n");
printf("Número de Page Faults: %u\n", pf);
printf("Número de Páginas Sujas Escritas: %u\n", pwSujas);
}

```

```

int main(int argc, char *argv[]) {
    if (argc != 5) {
        printf("Uso: %s <algoritmo> <arquivo.log> <tamanho_pagina_kb> "
               "<tamanho_memoria_mb>\n",
               argv[0]);
        return 1;
    }

    const char *alg = argv[1];
    const char *arqNome = argv[2];
    unsigned int pgTam_kb = atoi(argv[3]);
    unsigned int memTam_mb = atoi(argv[4]);

    simulador(alg, arqNome, pgTam_kb, memTam_mb);
    return 0;
}

```

Resumo do projeto:

O simulador apresentado consiste em um programa para avaliar o desempenho de diferentes algoritmos de substituição de páginas no contexto do gerenciamento de memória virtual. Ele recebe parâmetros como o algoritmo de substituição a ser utilizado, o arquivo de log com as referências de memória, o tamanho da página e o tamanho da memória física. A partir desses dados, o programa executa uma simulação passo a passo do carregamento e substituição de páginas na memória, contabilizando o número de page faults (faltas de página) e o número de páginas sujas escritas em disco. No final, um relatório é exibido, fornecendo métricas de desempenho que possibilitam comparar e analisar a eficácia de cada algoritmo.

A principal estrutura de dados utilizada é um array de Frame, cada elemento representando um quadro de memória física. Cada Frame contém o número da página atualmente carregada (pgNum), um bit indicando se a página foi recentemente referenciada (referenciada), um bit sinalizando se a página foi modificada (modificada), além de informações temporais, como o

momento do último acesso (ultAcesso) e o instante em que a página foi carregada (carregada). Esses dados permitem que os algoritmos de substituição selecionem quais páginas remover da memória de forma mais eficiente, baseando-se em padrões de acesso, antiguidade da referência, modificação e previsões futuras de uso.

O ciclo de simulação funciona da seguinte maneira: as referências de memória são lidas sequencialmente do arquivo de log. Para cada referência, calcula-se o número da página correspondente ao endereço acessado. Em seguida, verifica-se se essa página já está presente em um dos frames. Caso positivo, não ocorre uma falta de página; os bits e timestamps do frame são atualizados, marcando que a página foi novamente acessada e, caso necessário, que foi modificada. Caso a página não esteja presente, ocorre um page fault, incrementando o contador de faltas. Nesse momento, o simulador precisa escolher uma página para remover da memória, acionando um dos algoritmos de substituição. Ao remover uma página modificada, incrementa-se o contador de páginas sujas escritas no disco.

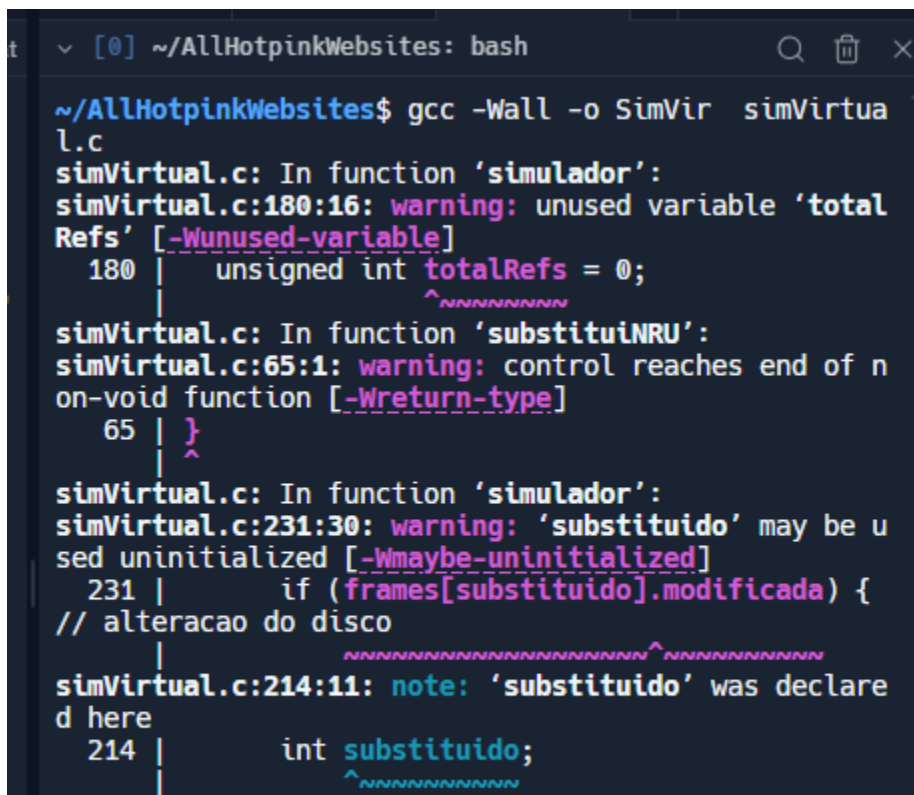
Cada algoritmo de substituição é implementado em uma função própria, refletindo critérios distintos para a escolha da página vítima:

- NRU: O algoritmo NRU divide as páginas em classes de acordo com seus bits de referência e modificação. Periodicamente (a cada quatro substituições), os bits de referência são zerados, garantindo que apenas as páginas realmente não utilizadas recentemente permaneçam marcadas como tal. Quando uma falta de página ocorre, o algoritmo prioriza remover páginas não referenciadas e não modificadas, pois elas provavelmente não serão necessárias imediatamente e não exigirão escrita em disco.
- LRU: O LRU se baseia na ideia de que a página acessada há mais tempo tem menor probabilidade de ser necessária novamente em breve. O simulador utiliza o campo ultAcesso para identificar qual página foi usada mais antigamente. Assim, em caso de falta de página, remove-se aquela cujo ultAcesso for o mais remoto, simulando a lógica de que as páginas mais antigas são as melhores candidatas a serem removidas.
- Clock: O algoritmo Clock faz uso de um ponteiro circular que percorre os frames. Ele verifica o bit de referência de cada página. Se encontrar uma página com o bit desligado, significa que a página não foi acessada recentemente, sendo uma boa candidata à remoção. Se o bit estiver ligado, o algoritmo apenas o desliga e avança o ponteiro, dando mais tempo para a página permanecer em memória. Esse processo é repetido até encontrar uma vítima cujo bit de referência esteja zerado. O Clock é uma implementação prática, simples e com menor sobrecarga quando comparada ao LRU.

- Second Chance: O Second Chance é semelhante ao Clock, porém incorpora a ideia de dar uma “segunda chance” às páginas, levando em conta não apenas o bit de referência, mas também o momento de carregamento da página (carregada). Se a página for considerada para remoção, mas estiver marcada como referenciada, ela tem seus bits e timestamps ajustados, recebendo outra oportunidade antes de ser finalmente removida. Isso adiciona um refinamento ao algoritmo Clock, equilibrando seu custo com uma seleção de vítimas mais inteligente.
- Ótimo (Optimal): O algoritmo Ótimo analisa as referências futuras, as quais são conhecidas em função do arquivo de log completo. Ele procura a página que ficará mais tempo sem ser acessada novamente e a remove, garantindo a melhor decisão possível naquele instante. Como o conhecimento futuro dos acessos não é realista em um ambiente operacional prático, este algoritmo serve principalmente como parâmetro teórico para avaliar a eficácia das demais abordagens. Na implementação, o simulador percorre as referências a partir da posição atual e encontra a página cujo próximo uso ocorrerá mais distante no tempo, selecionando-a para remoção.

Testes:

- Como compilar:



```

[0] ~/AllHotpinkWebsites: bash

~/AllHotpinkWebsites$ gcc -Wall -o SimVir simVirtual.c
simVirtual.c: In function 'simulador':
simVirtual.c:180:16: warning: unused variable 'total Refs' [-Wunused-variable]
  180 |     unsigned int totalRefs = 0;
      |                  ^~~~~~
simVirtual.c: In function 'substituiNRU':
simVirtual.c:65:1: warning: control reaches end of non-void function [-Wreturn-type]
   65 | }
      | ^
simVirtual.c: In function 'simulador':
simVirtual.c:231:30: warning: 'substituido' may be used uninitialized [-Wmaybe-uninitialized]
   231 |     if (frames[substituido].modificada) {
      |                      ~~~~~^~~~~~
// alteracao do disco
simVirtual.c:214:11: note: 'substituido' was declared here
   214 |     int substituido;
      |         ^~~~~~

```

- Usando **matriz.log**
8KB e 2MB:

```
Assistant  >_ Console  Shell  x  Integrations  +
v [0] ~/aaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaa$ ./sim Optimal matriz.log 8 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 2969
Número de Páginas Sujas Escritas: 1109
~/aaaaaaaaaaaaaaaaaaaa$ ./sim LRU matriz.log 8 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 4745
Número de Páginas Sujas Escritas: 1623
~/aaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance matriz.log 8
2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 6672
Número de Páginas Sujas Escritas: 2281
~/aaaaaaaaaaaaaaaaaaaa$ ./sim NRU matriz.log 8 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 248781
Número de Páginas Sujas Escritas: 33397
~/aaaaaaaaaaaaaaaaaaaa$
```

32KB e 1MB:

```
Assistant  >_ Console  Shell  ×  Integrations  +  
v [0] ~/aaaaaaaaaaaaaaaaaaaa: bash  
  
~/aaaaaaaaaaaaaaaaaaaa$ ./sim Optimal matriz.log 32 1  
Executando o simulador...  
Arquivo de entrada: matriz.log  
Tamanho da memoria fisica: 1MB  
Tamanho das páginas: 32KB  
Algoritmo de substituição: Optimal  
Relatório de Simulação:  
Número de Page Faults: 19115  
Número de Páginas Sujas Escritas: 4787  
~/aaaaaaaaaaaaaaaaaaaa$ ./sim LRU matriz.log 32 1  
Executando o simulador...  
Arquivo de entrada: matriz.log  
Tamanho da memoria fisica: 1MB  
Tamanho das páginas: 32KB  
Algoritmo de substituição: LRU  
Relatório de Simulação:  
Número de Page Faults: 32282  
Número de Páginas Sujas Escritas: 6582  
~/aaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance matriz.log 32 1  
Executando o simulador...  
Arquivo de entrada: matriz.log  
Tamanho da memoria fisica: 1MB  
Tamanho das páginas: 32KB  
Algoritmo de substituição: SecondChance  
Relatório de Simulação:  
Número de Page Faults: 59061  
Número de Páginas Sujas Escritas: 13543  
~/aaaaaaaaaaaaaaaaaaaa$ ./sim NRU matriz.log 32 1  
Executando o simulador...  
Arquivo de entrada: matriz.log  
Tamanho da memoria fisica: 1MB  
Tamanho das páginas: 32KB  
Algoritmo de substituição: NRU  
Relatório de Simulação:  
Número de Page Faults: 317455  
Número de Páginas Sujas Escritas: 44399  
~/aaaaaaaaaaaaaaaaaaaa$ █
```

8KB e 1MB:

```
Assistant  >_ Console  Shell  x  Integrations  +

v [0] ~/aaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaa$ ./sim Optimal matriz.log 8 1
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 5361
Número de Páginas Sujas Escritas: 1827
~/aaaaaaaaaaaaaaaaaaaa$ ./sim LRU matriz.log 8 1
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 10928
Número de Páginas Sujas Escritas: 3273
~/aaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance matriz.log 8 1
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 15410
Número de Páginas Sujas Escritas: 4505
~/aaaaaaaaaaaaaaaaaaaa$ ./sim NRU matriz.log 8 1
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 274461
Número de Páginas Sujas Escritas: 39309
~/aaaaaaaaaaaaaaaaaaaa$
```

32KB e 2MB:

```
Assistant  Console  Shell  Integrations  +
v [0] ~/aaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaa$ ./sim Optimal matriz.log 32 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 8008
Número de Páginas Sujas Escritas: 2618
~/aaaaaaaaaaaaaaaaaaaa$ ./sim LRU matriz.log 32 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 13230
Número de Páginas Sujas Escritas: 3877
~/aaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance matriz.log 32 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 19309
Número de Páginas Sujas Escritas: 5288
~/aaaaaaaaaaaaaaaaaaaa$ ./sim NRU matriz.log 32 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 287839
Número de Páginas Sujas Escritas: 41591
~/aaaaaaaaaaaaaaaaaaaa$
```

- Usando **compilador.log**
8KB e 2MB:

```
| ^
eriquita@batatatraste:~/Documents/SO$ ./sim Optimal compilador.log 8 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 10118
Número de Páginas Sujas Escritas: 2190
eriquita@batatatraste:~/Documents/SO$ ./sim LRU compilador.log 8 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 21091
Número de Páginas Sujas Escritas: 3839
eriquita@batatatraste:~/Documents/SO$ ./sim SecondChance compilador.log 8 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 26928
Número de Páginas Sujas Escritas: 5404
eriquita@batatatraste:~/Documents/SO$ ./sim NRU compilador.log 8 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 124685
Número de Páginas Sujas Escritas: 8565
eriquita@batatatraste:~/Documents/SO$
```

32KB e 1MB:

```
eriquita@batatatraste:~/Documents/SO$ ./sim Optimal compilador.log 32 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 38814
Número de Páginas Sujas Escritas: 6203
eriquita@batatatraste:~/Documents/SO$ ./sim LRU compilador.log 32 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 60336
Número de Páginas Sujas Escritas: 9429
eriquita@batatatraste:~/Documents/SO$ ./sim SecondChance compilador.log 32 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 72140
Número de Páginas Sujas Escritas: 13794
eriquita@batatatraste:~/Documents/SO$ ./sim NRU compilador.log 32 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 165740
Número de Páginas Sujas Escritas: 13642
eriquita@batatatraste:~/Documents/SO$
```


32KB e 2MB:

```
eriquita@batatatraste:~/Documents/S0$ ./sim Optimal compilador.log 32 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 23758
Número de Páginas Sujas Escritas: 3750
eriquita@batatatraste:~/Documents/S0$ ./sim LRU compilador.log 32 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 38641
Número de Páginas Sujas Escritas: 6186
eriquita@batatatraste:~/Documents/S0$ ./sim SecondChance compilador.log 32 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 47785
Número de Páginas Sujas Escritas: 9185
eriquita@batatatraste:~/Documents/S0$ ./sim NRU compilador.log 32 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 148943
Número de Páginas Sujas Escritas: 10463
eriquita@batatatraste:~/Documents/S0$
```

8KB e 1MB:

```
~ [0] ~/aaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaa$ ./ooo Optimal compilador.log 8 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 21271
Número de Páginas Sujas Escritas: 3711
~/aaaaaaaaaaaaaaaaaaaa$ ./ooo LRU compilador.log 8 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 36707
Número de Páginas Sujas Escritas: 5775
~/aaaaaaaaaaaaaaaaaaaa$ ./ooo SecondChance compilador.lo
g 8 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 44465
Número de Páginas Sujas Escritas: 8135
~/aaaaaaaaaaaaaaaaaaaa$ ./ooo NRU compilador.log 8 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 160632
Número de Páginas Sujas Escritas: 11851
~/aaaaaaaaaaaaaaaaaaaa$
```

- Usando **compressor.log**

8KB e 2MB:

```
Assistant >_ Console Shell x Integrations +
v [0] ~/aaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaa$ ./sim Optimal compressor.log 8 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 255
Número de Páginas Sujas Escritas: 0
~/aaaaaaaaaaaaaaaaaaaa$ ./sim LRU compressor.log 8 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 255
Número de Páginas Sujas Escritas: 0
~/aaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance compressor.log 8 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 255
Número de Páginas Sujas Escritas: 0
~/aaaaaaaaaaaaaaaaaaaa$ ./sim NRU compressor.log 8 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 255
Número de Páginas Sujas Escritas: 0
~/aaaaaaaaaaaaaaaaaaaa$ █
```

32KB e 1MB:

```
Assistant 7 Console Shell Integrations
v [0] ~/aaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaa$ ./sim Optimal compressor.log 32 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 607
Número de Páginas Sujas Escritas: 250
~/aaaaaaaaaaaaaaaaaaaa$ ./sim LRU compressor.log 32 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 911
Número de Páginas Sujas Escritas: 345
~/aaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance compressor.log 32 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 1119
Número de Páginas Sujas Escritas: 440
~/aaaaaaaaaaaaaaaaaaaa$ ./sim NRU compressor.log 32 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 65215
Número de Páginas Sujas Escritas: 6759
~/aaaaaaaaaaaaaaaaaaaa$ █
```

32KB e 2MB:

```

v [0] ~/aaaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaaa$ ./sim Optimal compressor.log 32 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memória física: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 334
Número de Páginas Sujas Escritas: 109
~/aaaaaaaaaaaaaaaaaaaaa$ ./sim LRU compressor.log 32 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memória física: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 515
Número de Páginas Sujas Escritas: 180
~/aaaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance compressor.log 32 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memória física: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 640
Número de Páginas Sujas Escritas: 257
~/aaaaaaaaaaaaaaaaaaaaa$ ./sim NRU compressor.log 32 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memória física: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 23726
Número de Páginas Sujas Escritas: 350
~/aaaaaaaaaaaaaaaaaaaaa$ █

```

8KB e 1MB:

```
Assistant  Console  Shell  Integrations  +
v [0] ~/aaaaaaaaaaaaaaaaaaaaa: bash

~/aaaaaaaaaaaaaaaaaaaaa$ ./sim Optimal compressor.log 8 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 345
Número de Páginas Sujas Escritas: 86
~/aaaaaaaaaaaaaaaaaaaaa$ ./sim LRU compressor.log 8 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 533
Número de Páginas Sujas Escritas: 132
~/aaaaaaaaaaaaaaaaaaaaa$ ./sim SecondChance compressor.log 8 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 570
Número de Páginas Sujas Escritas: 191
~/aaaaaaaaaaaaaaaaaaaaa$ ./sim NRU compressor.log 8 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 17641
Número de Páginas Sujas Escritas: 1
~/aaaaaaaaaaaaaaaaaaaaa$ █
```

- Usando **simulador.log**

8KB e 2MB:

```
eriquita@batatatriste:~/Documents/SO$ ./sim Optimal simulador.log 8 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 4748
Número de Páginas Sujas Escritas: 2160
eriquita@batatatriste:~/Documents/SO$ ./sim LRU simulador.log 8 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 8556
Número de Páginas Sujas Escritas: 3395
eriquita@batatatriste:~/Documents/SO$ ./sim SecondChance simulador.log 8 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 11884
Número de Páginas Sujas Escritas: 4655
eriquita@batatatriste:~/Documents/SO$ ./sim NRU simulador.log 8 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 185045
Número de Páginas Sujas Escritas: 23995
eriquita@batatatriste:~/Documents/SO$
```

32KB e 1MB:

```
eriquita@batatatraste:~/Documents/SO$ ./sim Optimal simulador.log 32 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 28328
Número de Páginas Sujas Escritas: 7581
eriquita@batatatraste:~/Documents/SO$ ./sim LRU simulador.log 32 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 48088
Número de Páginas Sujas Escritas: 10446
eriquita@batatatraste:~/Documents/SO$ ./sim SecondChance simulador.log 32 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 60922
Número de Páginas Sujas Escritas: 14653
eriquita@batatatraste:~/Documents/SO$ ./sim NRU simulador.log 32 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 187879
Número de Páginas Sujas Escritas: 34242
eriquita@batatatraste:~/Documents/SO$
```

32KB e 2MB:


```
eriquita@batatatraste:~/Documents/SO$ ./sim Optimal simulador.log 32 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 11870
Número de Páginas Sujas Escritas: 4421
eriquita@batatatraste:~/Documents/SO$ ./sim LRU simulador.log 32 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 22126
Número de Páginas Sujas Escritas: 7003
eriquita@batatatraste:~/Documents/SO$ ./sim SecondChance simulador.log 32 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 31700
Número de Páginas Sujas Escritas: 9201
eriquita@batatatraste:~/Documents/SO$ ./sim NRU simulador.log 32 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 183317
Número de Páginas Sujas Escritas: 32112
eriquita@batatatraste:~/Documents/SO$
```

8KB e 1MB:

```
eriquita@batatatraste:~/Documents/SO$ ./sim Optimal simulador.log 8 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Optimal
Relatório de Simulação:
Número de Page Faults: 9179
Número de Páginas Sujas Escritas: 3528
eriquita@batatatraste:~/Documents/SO$ ./sim LRU simulador.log 8 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: LRU
Relatório de Simulação:
Número de Page Faults: 16479
Número de Páginas Sujas Escritas: 5546
eriquita@batatatraste:~/Documents/SO$ ./sim SecondChance simulador.log 8 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: SecondChance
Relatório de Simulação:
Número de Page Faults: 22816
Número de Páginas Sujas Escritas: 7413
eriquita@batatatraste:~/Documents/SO$ ./sim NRU simulador.log 8 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: NRU
Relatório de Simulação:
Número de Page Faults: 207755
Número de Páginas Sujas Escritas: 34757
eriquita@batatatraste:~/Documents/SO$
```

Outros testes:

```
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compressor.log 8 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 255
Número de Páginas Sujas Escritas: 0
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compilador.log 8 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 22655
Número de Páginas Sujas Escritas: 4162
eriquita@batatatraste:~/Documents/SO$ ./sim Clock matriz.log 8 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 4870
Número de Páginas Sujas Escritas: 1676
eriquita@batatatraste:~/Documents/SO$ ./sim Clock simulador.log 8 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 9140
Número de Páginas Sujas Escritas: 3666
eriquita@batatatraste:~/Documents/SO$
```

```
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compressor.log 32 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 994
Número de Páginas Sujas Escritas: 379
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compilador.log 32 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 63302
Número de Páginas Sujas Escritas: 9988
eriquita@batatatraste:~/Documents/SO$ ./sim Clock simulador.log 32 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 50484
Número de Páginas Sujas Escritas: 11183
eriquita@batatatraste:~/Documents/SO$ ./sim Clock matriz.log 32 1
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 35656
Número de Páginas Sujas Escritas: 7527
eriquita@batatatraste:~/Documents/SO$
```

```
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compressor.log 32 2
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 537
Número de Páginas Sujas Escritas: 194
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compilador.log 32 2
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 40740
Número de Páginas Sujas Escritas: 6829
eriquita@batatatraste:~/Documents/SO$ ./sim Clock matriz.log 32 2
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 14331
Número de Páginas Sujas Escritas: 4120
eriquita@batatatraste:~/Documents/SO$ ./sim Clock simulador.log 32 2
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 2MB
Tamanho das páginas: 32KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 25666
Número de Páginas Sujas Escritas: 7647
eriquita@batatatraste:~/Documents/SO$
```

```
eriquita@batatatraste:~/Documents/SO$ ./sim Clock simulador.log 8 1
Executando o simulador...
Arquivo de entrada: simulador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 17923
Número de Páginas Sujas Escritas: 6070
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compressor.log 8 1
Executando o simulador...
Arquivo de entrada: compressor.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 577
Número de Páginas Sujas Escritas: 163
eriquita@batatatraste:~/Documents/SO$ ./sim Clock compilador.log 8 1
Executando o simulador...
Arquivo de entrada: compilador.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 38827
Número de Páginas Sujas Escritas: 6250
eriquita@batatatraste:~/Documents/SO$ ./sim Clock matriz.log 8 1
Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria fisica: 1MB
Tamanho das páginas: 8KB
Algoritmo de substituição: Clock
Relatório de Simulação:
Número de Page Faults: 12896
Número de Páginas Sujas Escritas: 3844
eriquita@batatatraste:~/Documents/SO$
```

Descrição algoritmo ótimo:

Nós conseguimos fazer a implementação do algoritmo ótimo, segue uma breve explicação de como implementamos (a implementação está no nosso código enviado no ead):

Ele é implementado levando em consideração que a sequência completa de acessos futuros está disponível ao simulador. Com base nessa informação, o algoritmo escolhe para remoção a página cujo próximo acesso está mais distante no tempo, ou que sequer será acessada novamente, garantindo assim a decisão mais eficiente possível em termos teóricos.

No código, a implementação do algoritmo Ótimo começa recebendo como parâmetros o número de frames (`framesNum`), o array de referências futuras (`referencias`), o total de referências (`totalRefs`) e a posição atual de acesso (`posAtual`). A partir daí, o algoritmo percorre cada frame carregado na memória para identificar a próxima vez em que a página de cada frame será acessada. Isso é feito examinando o array de referências da posição atual em diante. Se uma página não for encontrada nos acessos futuros, ela se torna imediatamente a melhor candidata a ser removida, já que não haverá necessidade de tê-la em memória novamente.

Caso todas as páginas analisadas tenham um acesso futuro, o algoritmo identifica qual delas será utilizada mais adiante no tempo. Ele rastreia a distância entre a posição atual e o próximo acesso de cada página, selecionando aquela cujo próximo uso está mais distante. Essa página é considerada a escolha ideal, já que liberar seu espaço agora minimiza a probabilidade de um acesso iminente.

Pseudo-Código:

`maisDistante = 0`

`substituido = -1`

Loop para verificar se há frame vazio e caso haja substitui esse

Loop para cada frame:

`pgNumAtual = pagina do frame`

`flag encontrada inicialmente falsa`

`distancia inicialmente zero`

Loop para ver as referências começando da posição atual:

`se a referência for igual a página atual:`

`flag encontrada é verdadeira`

`distancia é calculada`

`loop quebrado`

Se não for encontrada retorna a página(substitui aquela que não tem mais referências)

Se a distancia calculada for maior que a maior distância:
maior distância atualizada
melhor candidato de substituição é tal página

retorna o candidato

Análise do desempenho dos algoritmos:

Na maioria das análises, reduzir o tamanho da página (por exemplo, de 32KB para 8KB) diminui o número de page faults e de páginas sujas, melhorando a eficiência dos algoritmos, exceto em algumas situações envolvendo o NRU. Quando o número de page faults é menor ou igual ao número de frames disponíveis, não há necessidade de efetuar substituições de páginas (page out). Nesses casos, todos os algoritmos tendem a apresentar resultados idênticos.

Um exemplo é a análise do arquivo "compressor 8 2", na qual não houve substituições efetivas devido à ausência de necessidade de troca de páginas, resultando em métricas idênticas para todos os algoritmos.

Análise do matriz.log

- Memória: 2MB:
 - Com páginas de 8KB: A ordem de eficiência para page faults e páginas sujas é Ótimo, LRU, Second Chance, NRU.
 - Com páginas de 32KB: A mesma ordem se mantém tanto para page faults quanto para páginas sujas.
Observa-se que, com páginas menores (8KB), os valores absolutos de page faults e páginas sujas são significativamente menores.
- Memória: 1MB:
 - Com 32KB e 8KB, a ordem dos algoritmos em relação a page faults e páginas sujas permanece a mesma (Ótimo, LRU, Second Chance, NRU), porém com menor número absoluto de falhas e páginas sujas no caso das páginas de 8KB.

Análise do compilador.log

- Memória: 2MB / Páginas: 8KB:
A ordem permanece: Ótimo, LRU, Second Chance, NRU, tanto para page faults quanto para páginas sujas, com números absolutos menores do que com páginas maiores.
- Memória: 2MB / Páginas: 32KB:
A ordem permanece inalterada em termos de page faults, porém a hierarquia de

páginas sujas segue a mesma lógica (Ótimo, LRU, Second Chance, NRU).

Observa-se que a mudança no tamanho da página afeta quantitativamente os resultados, mas não altera a ordem dos algoritmos.

- Memória: 1MB / Páginas: 32KB:
Page faults: Ótimo, LRU, Second Chance, NRU
Páginas sujas: Ótimo, LRU, NRU, Second Chance
 - Memória: 1MB / Páginas: 8KB:
Page faults: Ótimo, LRU, Second Chance, NRU
Páginas sujas: Ótimo, LRU, Second Chance, NRU
Novamente, números absolutos menores com páginas de 8KB.
-

Análise do simulador.log

- Memória: 2MB / Páginas: 8KB e 32KB:
A ordem permanece a mesma tanto para page faults quanto para páginas sujas (Ótimo, LRU, Second Chance, NRU). No caso de páginas de 8KB, obtêm-se valores mais baixos para page faults e páginas sujas, com exceção do NRU.
 - Memória: 1MB / Páginas: 8KB e 32KB:
A ordem dos algoritmos não se altera, permanecendo Ótimo, LRU, Second Chance, NRU, tanto para page faults quanto para páginas sujas.
Com páginas menores, novamente obtêm-se menores quantidades absolutas de page faults e páginas sujas, exceto para o NRU.
-

Análise do compressor.log

- Memória: 2MB / Páginas: 8KB:
Não houve necessidade de substituições, resultando em igualdade nos resultados de todos os algoritmos, devido ao número de page faults ser inferior ao número de frames.
- Memória: 2MB / Páginas: 32KB:
A ordem retorna a ser: Ótimo, LRU, Second Chance, NRU tanto para page faults quanto para páginas sujas.
Com páginas menores, reduz-se a quantidade absoluta de page faults e páginas sujas.
- Memória: 1MB / Páginas: 8KB:
Page faults: Ótimo, LRU, Second Chance, NRU
Páginas sujas: NRU, Ótimo, LRU, Second Chance
Aqui, o NRU apresenta um resultado atípico, obtendo menos páginas sujas do que o Ótimo.
- Memória: 1MB / Páginas: 32KB:
Page faults e páginas sujas: Ótimo, LRU, Second Chance, NRU
Ao aumentar o tamanho da página, retoma-se a ordem padrão.

A eficiência do NRU é altamente dependente da frequência com que o bit de referência é resetado, podendo modificar seu desempenho em termos de pages sujas e page faults, tornando-o mais sensível aos ajustes de parâmetros. De forma geral, o algoritmo Ótimo apresenta os melhores resultados, seguido do LRU, depois do Second Chance e, por último, do NRU. Todavia, há casos específicos em que o NRU pode apresentar menos páginas sujas que o Ótimo, e situações em que todos os algoritmos exibem resultados equivalentes devido à ausência de trocas efetivas de páginas. Além disso, a escolha do tamanho da página e da quantidade de memória tem impacto significativo nos resultados, sendo que páginas menores (8KB) costumam reduzir tanto page faults quanto páginas sujas, exceto no caso do NRU, que é mais sensível à frequência de reinicialização do bit de referência.