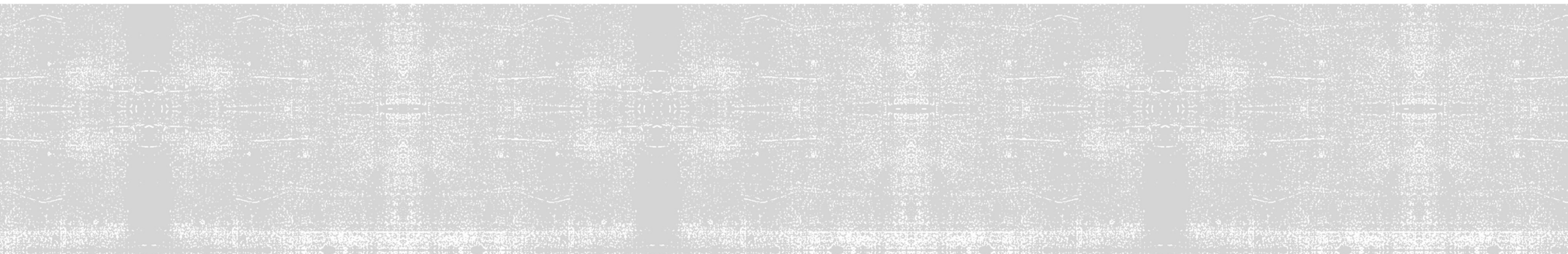




DETTAGLI IMPLEMENTATIVI



PROBLEMI DI IMPLEMENTAZIONE DELLA MEMORIA VIRTUALE

- **Sfide nell'Implementazione:**

- Scelta tra algoritmi teorici (es. Seconda Chance, Aging) e pratiche operative (allocazione locale/globale, paginazione a richiesta/prepaginazione).
- Gestione di problemi pratici di implementazione della memoria virtuale.

- **Attività del Sistema Operativo nella Paginazione:**

- **Creazione del Processo:**

- Determinare le dimensioni iniziali del programma e dei dati.
- Creare e inizializzare la tabella delle pagine.
- Allocare spazio nella memoria non volatile per lo scambio.
- Inizializzare l'area di scambio e registrare informazioni nella tabella dei processi.

- **Esecuzione del Processo:**

- Azzerrare la MMU e, se necessario, svuotare il TLB.
- Rendere attiva la tabella delle pagine del processo.
- **Pre-paginazione:** Facoltativamente, caricare alcune pagine in memoria per ridurre i page fault iniziali.



GESTIONE DEI PAGE FAULT E CHIUSURA DEL PROCESSO

- **Gestione dei Page Fault:**
 - **Determinare l'indirizzo virtuale** che ha causato il **fault**.
 - **Trovare la pagina necessaria** nella memoria non volatile.
 - **Scegliere un frame disponibile**, eventualmente sfrattando pagine vecchie.
 - **Caricare la pagina nel frame** e ripristinare il contatore del programma.
- **Chiusura del Processo:**
 - **Rilasciare la tabella delle pagine**, le pagine in memoria e lo spazio su disco/SSD.
 - **Gestire le pagine condivise con altri processi**, rilasciandole solo dopo l'ultimo utilizzo.



PAGE FAULT IN 10 PASSI:

A. INIZIO DELLA SEQUENZA

1. **Trap nel Kernel da Parte dell'Hardware:**

- L'hardware esegue una trap nel kernel, salvando il contatore del programma nello stack.
- Informazioni sull'istruzione corrente salvate nei registri speciali della CPU.

2. **Avvio Routine di Servizio Interrupt:**

- Viene eseguita una routine in assembly per salvare i registri e altre informazioni volatili.
- Invocazione del gestore dei page fault.

3. **Identificazione della Pagina Virtuale Necessaria:**

- Il sistema operativo determina quale pagina virtuale manca.
- Se non disponibile dai registri hardware, recupero e analisi dell'istruzione dal contatore di programma.



PAGE FAULT IN 10 PASSI

B. GESTIONE E RISOLUZIONE

4. **Verifica Validità Indirizzo e Protezione:**

- Controllo della validità dell'indirizzo e coerenza della protezione con l'accesso.
- Se invalide, invio di un segnale di errore o terminazione del processo.

5. **Rilascio di un Frame Libero:**

- Se non ci sono frame liberi, esecuzione di un algoritmo di sostituzione delle pagine.
- Se la pagina è "sporca", viene schedulata per la scrittura in memoria non volatile e il processo è sospeso.

6. **Caricamento della Pagina Richiesta:**

- Una volta liberato (o scritto in memoria non volatile), il frame viene usato per caricare la pagina necessaria da disco o SSD.
- Durante il caricamento della pagina, il processo in page fault è ancora sospeso e viene eseguito, se disponibile, un altro processo utente.



PAGE FAULT IN 10 PASSI

C. CONCLUSIONE E RIPRESA

7. **Aggiornamento delle Tabelle delle Pagine:**

- Al completamento del trasferimento dal supporto non volatile, le tabelle delle pagine vengono aggiornate per riflettere la nuova posizione della pagina.
- Il frame viene contrassegnato come disponibile.

8. **Ripristino dell'Istruzione in Errore:**

- L'istruzione in errore è riportata allo stato che aveva all'inizio
- Il contatore di programma è ripristinato in modo da puntare a quell'istruzione.

9. **Ripresa del Processo in Errore:**

- Il processo precedentemente in errore viene schedato per l'esecuzione.
- Ritorno alla routine in assembly che lo aveva interrotto.

10. **Ricarica dei Registri e Ritorno allo Spazio Utente:**

- La routine di servizio ricarica i registri e le informazioni di stato.
- Il controllo ritorna allo spazio utente per continuare l'esecuzione da dove era stata interrotta.



BLOCCARE LE PAGINE IN MEMORIA DURANTE L'I/O

Scenario:

- Un processo invia una richiesta di lettura da un file o dispositivo in un buffer nel suo spazio di indirizzi.
- Mentre attende il completamento dell'I/O, può essere sospeso per permettere l'esecuzione di un altro processo.

Problema con Page Fault

- Se il secondo processo genera un page fault, esiste il rischio che la pagina contenente il buffer di I/O venga selezionata per essere rimossa
- Se avviene un trasferimento DMA (Direct Memory Access) su quella pagina, la rimozione potrebbe causare scritture errate nei dati.

Soluzione: *Pinning delle Pagine*

- Le pagine utilizzate per l'I/O vengono "bloccate" o "pinned" (fissate) in memoria, prevenendo la loro rimozione.
- Questo approccio assicura che le operazioni di I/O possano procedere senza interruzioni.

Alternativa: Gestione I/O nei Buffer del Kernel

- Un'altra strategia è gestire l'I/O nei buffer del kernel e poi copiare i dati nelle pagine utente.
- Questo metodo richiede una copia aggiuntiva dei dati, potenzialmente rallentando il processo.



MEMORIA SECONDARIA E GESTIONE DELLO SCAMBIO

Ma dove viene messa una pagina quando viene spostata nella memoria non volatile dopo essere stata «paginata fuori» dalla memoria?

- **Gestione dello Spazio di Scambio (*file o partizione di swap*)**
 - Il sistema operativo prevede **una partizione speciale o dispositivo separato per lo scambio**, come nei sistemi UNIX.
 - **Un'area del disco/SSD strutturata in maniera differente dal file system** usato per memorizzare file e cartelle (vedi lezioni successive)
 - Partizione di scambio **con file system semplificato**, utilizzando numeri di blocchi relativi
- **Allocazione in Memoria di Scambio:**
 - All'avvio, allocazione di spazio in partizione di scambio pari alla dimensione del processo.
 - Gestione come lista di parti libere (anche se esistono miglioramenti, vedi Capitolo 10 del libro).
- **Associazione Processo-Area di Scambio:**
 - Ogni **processo ha un'area di scambio in memoria non volatile**.
 - L'indirizzo in cui scrivere una pagina è calcolato sommando l'offset della pagina al suo spazio virtuale all'inizio dell'area di scambio.



STRATEGIE DI PAGINAZIONE E OTTIMIZZAZIONI

- **Gestione di Crescita dei Processi:**

- Riserva di aree separate per testo, dati e stack, per gestire l'espansione dei processi.

- **Alternativa di Allocazione Dinamica:**

- Allocazione dello spazio su disco/SSD al momento dello scambio di ogni pagina.
- Tavola per ogni processo che indica la posizione di ogni pagina in memoria non volatile.

- **Esempi di Gestione Paginazione:**

- a) Paginazione in area di scambio statica. Ogni pagina ha una posizione fissa su disco.
- b) Salvataggio dinamico delle pagine. Indirizzo su disco scelto al momento dello scambio.

- **Ottimizzazioni su File System:**

- Uso di file pre-allocati in file system normale (es. Windows).

