Address Transformation

Generated on Mon Dec 23 2024 16:37:05 for Address Transformation by Doxygen 1.12.0

Mon Dec 23 2024 16:37:05

Chapter 1

File Index

1.1 File List

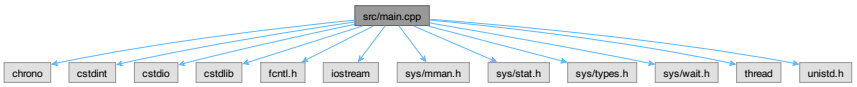
Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 src/main.cpp File Reference

#include <chrono>
#include <cstdint>
#include <cstdio>
#include <cstdlib>
#include <fcntl.h>
#include <iostream>
#include <sys/mman.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <thread>
#include <unistd.h>
Include dependency graph for main.cpp:



Functions

- uintptr_t virtual_to_physical_address (uintptr_t vaddr)

  Converts a virtual address to a physical address by reading /proc/self/pagemap.
- void verify_physical_address (const volatile int *turn, int i)

  Verifies that the physical address contents match the logical value of 'turn'.
- void peterson_algorithm (int i, volatile int *flag, volatile int *turn)

  The main routine for each process in Peterson's Algorithm.
- int main ()

  Program entry point.

2.1.1 Function Documentation

2.1.1.1 main()

int main ()

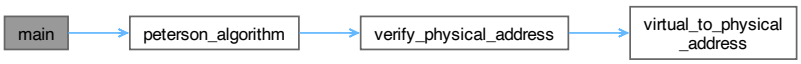
Program entry point.

Allocates shared memory for the Peterson Algorithm's flags and turn variable. Forks a child process, assigning roles (0 for parent, 1 for child). Both processes execute peterson_algorithm(). The parent waits for the child to finish and then releases the shared memory.

Returns

EXIT_SUCCESS on successful execution, or EXIT_FAILURE on errors.

Here is the call graph for this function:



2.1.1.2 peterson_algorithm()

void peterson_algorithm (
int i,
volatile int * flag,
volatile int * turn)

The main routine for each process in Peterson's Algorithm.

This function implements two iterations of the following:

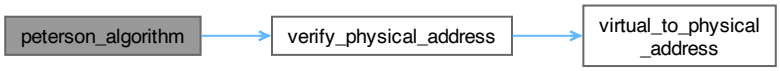
- Set the flag to indicate that this process wants to enter the critical section.
- Set turn to the other process.
- Busy-wait until it is safe to enter the critical section.
- Enter the critical section, verify the physical address mapping of 'turn'.
- Sleep for a moment to simulate some work in the critical section.
- Exit the critical section, reset the flag.
- Sleep to yield time to the other process.

Parameters

i	The process identifier (0 for parent, 1 for child).
flag	Pointer to the array of two flags (flag[0], flag[1]).

turn	Pointer to the shared 'turn' variable.

Here is the call graph for this function:



Here is the caller graph for this function:

2.1.1.3 verify_physical_address()

```
void verify_physical_address (
                const volatile int * turn,
                int i)
```

Verifies that the physical address contents match the logical value of 'turn'.

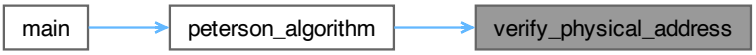
This function prints both the logical (virtual) and physical addresses of 'turn'. It then attempts to read the value from /dev/mem at the physical address to confirm that it matches the current value of 'turn'. Note that this step requires privileges to open /dev/mem.

Parameters

turn	Pointer to the shared 'turn' variable.
i	The process identifier (0 for parent, 1 for child).

Here is the call graph for this function:

Here is the caller graph for this function:



2.1.1.4 virtual_to_physical_address()

```
uintptr_t virtual_to_physical_address (
                uintptr_t vaddr)
```

Converts a virtual address to a physical address by reading /proc/self/pagemap.

This function reads the pagemap entry corresponding to the given virtual address, extracts the Page Frame Number (PFN) if present, and constructs the physical address.

Parameters

vaddr	The virtual address to be converted.

Returns

The corresponding physical address, or 0 on failure.

Here is the caller graph for this function:

Index