Mem. Mgmt: Segmentation

Each function of program is allocated memory separately.

No longer Contiguous

Dynamic Address Translation

Program generates address of the form

\[(S, O) \rightarrow \text{Seg} #, \text{offset}\]

\[\downarrow\]

Absolute address

Need a segment table (one / process)

\[\downarrow\]

Inside PCB

Use \( S \) as an index into segment table, get the entry (size, starting address)

**DAT**: if \( 0 \leq \text{size} \) (absolute address)

\[= SA + O\]
Segmentation reduces the severity of fragmentation; it does not eliminate fragmentation.

\[ \text{IV} \quad \text{Paging} \]

Page, size = m words

User Program

Program generates effective address \( \alpha \)

Convert \( \alpha \) into \((P, 0)\)

\[ p = \left\lfloor \frac{\alpha}{m} \right\rfloor; \quad 0 = \alpha \mod m \]

\( m = 2^k \) = Page Size; \( k \) is an int

\[ \begin{array}{c|c}
\text{Page \#} & \text{offset} \\
\hline
\text{address \( \alpha \)}
\end{array} \]
Dynamic Address Translation (DAT)

We need a table.

Page map Table

DATA:  P | O

effective
addres

Use P as an index into the page map table. Find P, the frame number where page P of our program resides in physical memory.

Concatenate f and O, we have absolute address.
Where do we store Page Map Table?

(i) in registers.
   - good: fast
   - bad: expensive

(ii) in PCB (main memory)
   - good: no extra cost
   - bad: slow

(iii) Cache/Translation Lookaside Buffer (TLB)

Hybrid Scheme

Store few entries in reg registers

Each entry: \( [p, f] \)

For DAS:

\[ \begin{array}{c}
p' \\
0' \\
\end{array} \]

effective address

We have high hit ratio
How do we determine page size?

\[ \text{Page Size} = 2^k \text{ words} \]

\( k \) is an int

↓, too small? Page map table is very large.

↑ too large? \underline{internal fragmentation}?

\[ \frac{X}{P} \] last page of a program may not be full

\( P \) is page size (words)

\( S \) is average program size (words)

\( X = \# \text{ of words needed to store one entry in page map table} \)

memory wasted per program

\[ W = \frac{P}{2} + \frac{S}{P} \cdot X \]

Find the value of \( P \) such that

\( W \) is minimized
\[
\frac{dw}{dp} = \frac{1}{2} - \frac{5x}{p^2} = 0
\]

\[p^2 = 25x; \quad p = \pm \sqrt{25x}\]

\[p = \sqrt{25x}\]

\[
\frac{d^2w}{dp^2} = \frac{5x}{p^3} \quad \text{is positive} \quad p = \sqrt{25x}
\]

\[\Rightarrow w \text{ is minimized when } p = \sqrt{25x}\]
Chapter 8 - Virtual Memory

Demand Paging

Store complete program (all pages) in hard drive

Store some pages of the program in physical memory

File Map Table

Disk address of page i of this program

Page map table entry of page j of this program

Page Map Table

Page #

Use p as an index into page map table. Find frame #f and valid bit.
if (valid bit is 1) {
    replace p by f;
    done
}

else { // we have page fault
    use p as an index into file map table; find disk address
    initiate data transfer from
    hard drive to memory (page in)
    blocks till page appears
    in memory;
    do the DAT;