

Quincy Flint

Virtual Memory

EEL 3713C: Digital Computer Architecture

Quincy Flint

[Ionospheric Radio Lab in NEB]

Outline

Quincy Flint

1. Memory Problems

- Not enough memory
- Holes in address space
- Programs overwriting

2. What is Virtual Memory?

- Layer of indirection
- How does indirection solve above
- Page tables and translation

3. How do we implement VM?

- Create and store page tables
- Fast address translation

4. Virtual Memory and Caches

- Prevent cache performance degradation when using VM

Quincy Flint

Multi-Level Page Tables

Quincy Flint

Page Table Size... again

Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...

Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - $32 \text{ bits (machine)} - 12 \text{ bits (page offset)} = 20 \text{ bits} = 1 \text{ million}$

Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits

Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - $32 \text{ bits (machine)} - 12 \text{ bits (page offset)} = 20 \text{ bits} = 1 \text{ million}$
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits

1M entries x 4 Bytes per entry
= 4 MB page tables

Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits
- BUT, each program needs its own page table

1M entries x 4 Bytes per entry
= 4 MB page tables

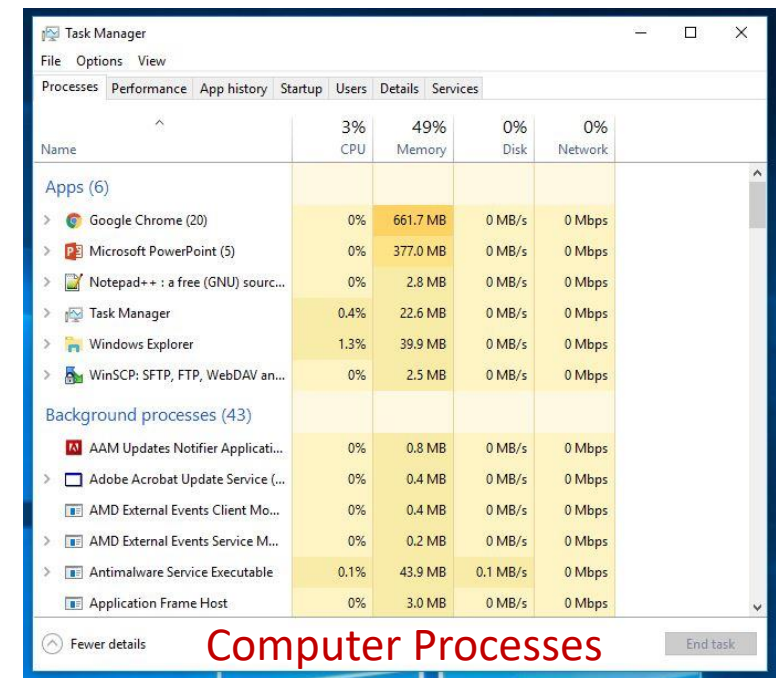
Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits

1M entries x 4 Bytes per entry
= 4 MB page tables

- BUT, each program needs its own page table



Quincy Flint

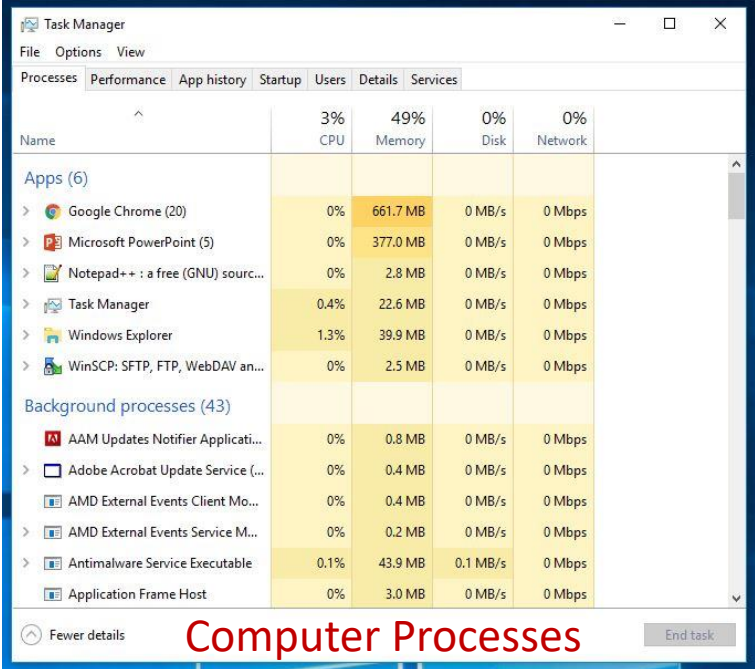
Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits

1M entries x 4 Bytes per entry
= **4 MB page tables**

4 MB page tables x 50 processes
= **200 MB of RAM for PTs**

- BUT, each program needs its own page table



The screenshot shows the Windows Task Manager Performance tab, specifically the 'Processes' section. It displays a table of running processes with columns for Name, CPU usage, Memory usage, Disk usage, and Network usage. The 'Memory' column is highlighted in yellow. The processes are grouped into 'Apps (6)' and 'Background processes (43)'. The total memory usage for all processes is 49%.

Name	CPU	Memory	Disk	Network
Apps (6)				
Google Chrome (20)	0%	661.7 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (5)	0%	377.0 MB	0 MB/s	0 Mbps
Notepad++ : a free (GNU) sourc...	0%	2.8 MB	0 MB/s	0 Mbps
Task Manager	0.4%	22.6 MB	0 MB/s	0 Mbps
Windows Explorer	1.3%	39.9 MB	0 MB/s	0 Mbps
WinSCP: SFTP, FTP, WebDAV an...	0%	2.5 MB	0 MB/s	0 Mbps
Background processes (43)				
AAM Updates Notifier Applicati...	0%	0.8 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...)	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Service M...	0%	0.2 MB	0 MB/s	0 Mbps
Antimalware Service Executable	0.1%	43.9 MB	0.1 MB/s	0 Mbps
Application Frame Host	0%	3.0 MB	0 MB/s	0 Mbps

Computer Processes

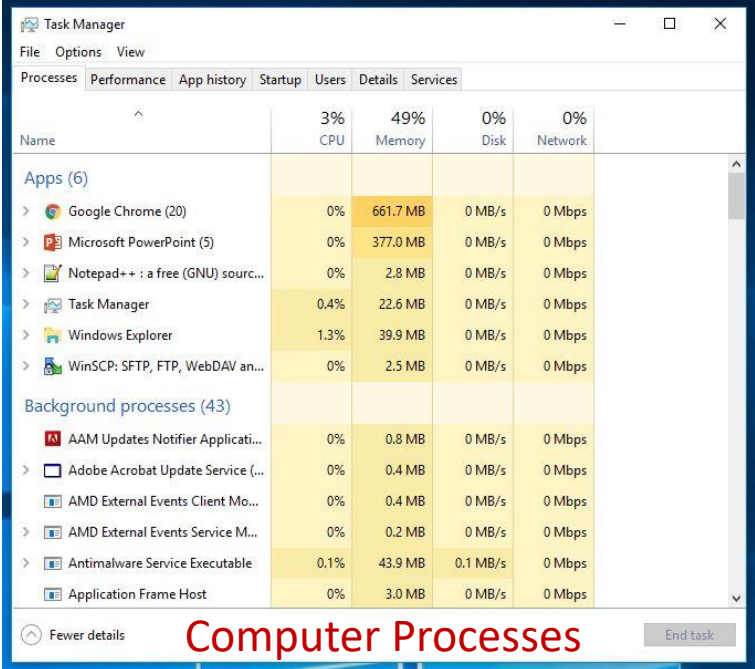
Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits
- BUT, each program needs its own page table
 - 100 programs require 400 MB of RAM

1M entries x 4 Bytes per entry
= **4 MB page tables**

4 MB page tables x 50 processes
= **200 MB of RAM for PTs**



The screenshot shows the Windows Task Manager Performance tab, specifically the 'Processes' section. It displays a table of running processes with columns for Name, CPU usage, Memory usage, Disk usage, and Network usage. The 'Memory' column is highlighted in yellow, and the total memory usage is shown as 49%.

Name	CPU	Memory	Disk	Network
Apps (6)				
Google Chrome (20)	0%	661.7 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (5)	0%	377.0 MB	0 MB/s	0 Mbps
Notepad++ : a free (GNU) sourc...	0%	2.8 MB	0 MB/s	0 Mbps
Task Manager	0.4%	22.6 MB	0 MB/s	0 Mbps
Windows Explorer	1.3%	39.9 MB	0 MB/s	0 Mbps
WinSCP: SFTP, FTP, WebDAV an...	0%	2.5 MB	0 MB/s	0 Mbps
Background processes (43)				
AAM Updates Notifier Applicati...	0%	0.8 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...)	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Service M...	0%	0.2 MB	0 MB/s	0 Mbps
Antimalware Service Executable	0.1%	43.9 MB	0.1 MB/s	0 Mbps
Application Frame Host	0%	3.0 MB	0 MB/s	0 Mbps

Computer Processes

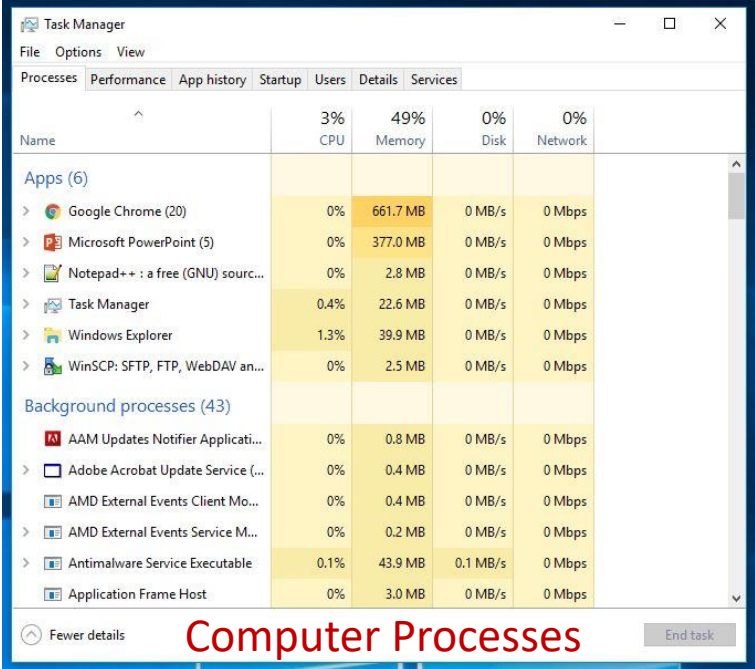
Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits
- BUT, each program needs its own page table
 - 100 programs require 400 MB of RAM
 - Cannot swap pages tables to disk (normally)

1M entries x 4 Bytes per entry
= **4 MB page tables**

4 MB page tables x 50 processes
= **200 MB of RAM for PTs**



The screenshot shows the Windows Task Manager Performance tab, specifically the 'Processes' section. It displays a table of running processes with columns for Name, CPU usage, Memory usage, Disk usage, and Network usage. The 'Memory' column is highlighted in yellow. The processes are grouped into 'Apps (6)' and 'Background processes (43)'. The total memory usage for all processes is 49%.

Name	CPU	Memory	Disk	Network
Apps (6)				
Google Chrome (20)	0%	661.7 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (5)	0%	377.0 MB	0 MB/s	0 Mbps
Notepad++ : a free (GNU) sourc...	0%	2.8 MB	0 MB/s	0 Mbps
Task Manager	0.4%	22.6 MB	0 MB/s	0 Mbps
Windows Explorer	1.3%	39.9 MB	0 MB/s	0 Mbps
WinSCP: SFTP, FTP, WebDAV an...	0%	2.5 MB	0 MB/s	0 Mbps
Background processes (43)				
AAM Updates Notifier Applicati...	0%	0.8 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...)	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Service M...	0%	0.2 MB	0 MB/s	0 Mbps
Antimalware Service Executable	0.1%	43.9 MB	0.1 MB/s	0 Mbps
Application Frame Host	0%	3.0 MB	0 MB/s	0 Mbps

Computer Processes

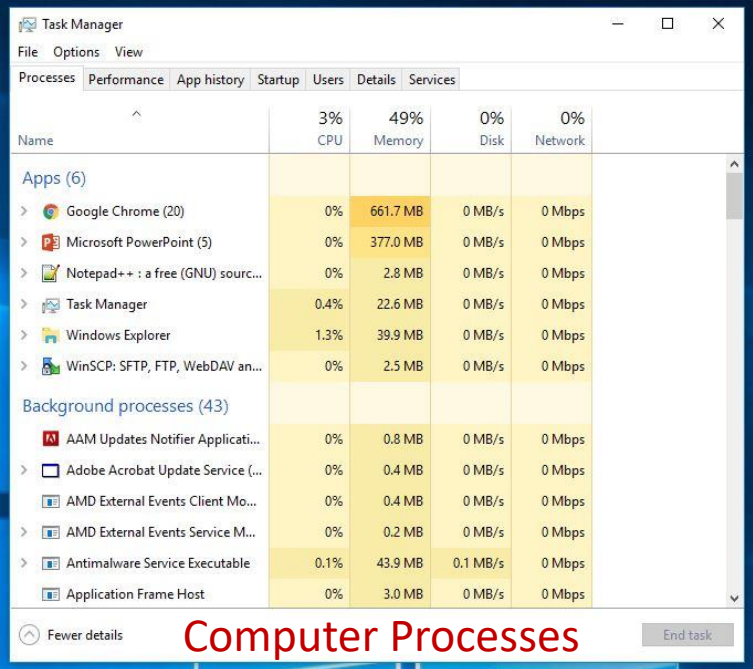
Quincy Flint

Page Table Size... again

- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits
- BUT, each program needs its own page table
 - 100 programs require 400 MB of RAM
 - Cannot swap pages tables to disk (normally)
- How can we fix this?

1M entries x 4 Bytes per entry
= **4 MB page tables**

4 MB page tables x 50 processes
= **200 MB of RAM for PTs**



The screenshot shows the Windows Task Manager Performance tab, specifically the 'Processes' section. It displays a table of running processes with columns for Name, CPU usage, Memory usage, Disk usage, and Network usage. The 'Memory' column is highlighted in yellow, and the total memory usage is shown as 49%.

Name	CPU	Memory	Disk	Network
Apps (6)				
Google Chrome (20)	0%	661.7 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (5)	0%	377.0 MB	0 MB/s	0 Mbps
Notepad++ : a free (GNU) sourc...	0%	2.8 MB	0 MB/s	0 Mbps
Task Manager	0.4%	22.6 MB	0 MB/s	0 Mbps
Windows Explorer	1.3%	39.9 MB	0 MB/s	0 Mbps
WinSCP: SFTP, FTP, WebDAV an...	0%	2.5 MB	0 MB/s	0 Mbps
Background processes (43)				
AAM Updates Notifier Applicati...	0%	0.8 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...)	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Service M...	0%	0.2 MB	0 MB/s	0 Mbps
Antimalware Service Executable	0.1%	43.9 MB	0.1 MB/s	0 Mbps
Application Frame Host	0%	3.0 MB	0 MB/s	0 Mbps

Computer Processes

Quincy Flint

Page Table Size... again

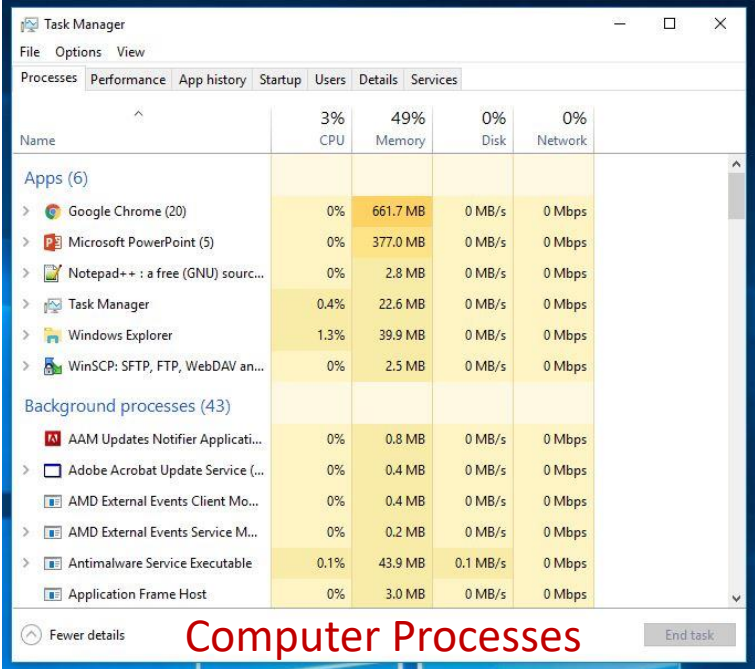
- Given a 32-bit machine, 4 kB pages...
 - 1 million PTEs
 - 32 bits (machine) – 12 bits (page offset) = 20 bits = 1 million
 - Each PTE is around 4 Bytes wide
 - 20 bits (Physical Page) + permission bits

- BUT, each program needs its own page table
 - 100 programs require 400 MB of RAM
 - Cannot swap pages tables to disk (normally)

- How can we fix this? Indirection!

1M entries x 4 Bytes per entry
= **4 MB page tables**

4 MB page tables x 50 processes
= **200 MB of RAM for PTs**



The screenshot shows the Windows Task Manager Performance tab, specifically the 'Memory' section. It displays a table of processes and their memory usage. The 'Memory' column is highlighted in yellow. The total system memory usage is 49%.

Name	CPU	Memory	Disk	Network
Apps (6)				
Google Chrome (20)	0%	661.7 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (5)	0%	377.0 MB	0 MB/s	0 Mbps
Notepad++ : a free (GNU) sourc...	0%	2.8 MB	0 MB/s	0 Mbps
Task Manager	0.4%	22.6 MB	0 MB/s	0 Mbps
Windows Explorer	1.3%	39.9 MB	0 MB/s	0 Mbps
WinSCP: SFTP, FTP, WebDAV an...	0%	2.5 MB	0 MB/s	0 Mbps
Background processes (43)				
AAM Updates Notifier Applicati...	0%	0.8 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...)	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...	0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Service M...	0%	0.2 MB	0 MB/s	0 Mbps
Antimalware Service Executable	0.1%	43.9 MB	0.1 MB/s	0 Mbps
Application Frame Host	0%	3.0 MB	0 MB/s	0 Mbps

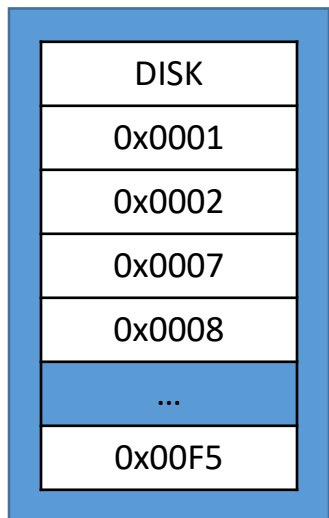
Quincy Flint

Multi-Level Page Tables

1st Level Page Table

4kB [1,024 entries]

Size of 1 Page



A vertical table with a blue border and a white background, representing a 1st level page table. It contains several rows, each representing a page entry. The entries are: 'DISK', '0x0001', '0x0002', '0x0007', '0x0008', an ellipsis '...', and '0x00F5'. The row containing the ellipsis is highlighted with a blue background.

DISK
0x0001
0x0002
0x0007
0x0008
...
0x00F5

Main Memory



A large, vertical green rectangle representing the main memory.

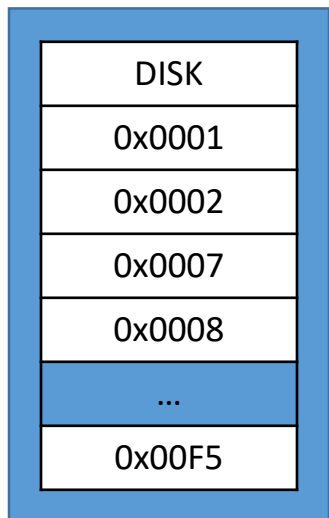
Quincy Flint

Multi-Level Page Tables

1st Level Page Table

4kB [1,024 entries]

Size of 1 Page



A vertical table with a blue border and white background, representing a 1st level page table. It contains the following entries from top to bottom: DISK, 0x0001, 0x0002, 0x0007, 0x0008, an ellipsis (...), and 0x00F5. The entry containing the ellipsis is highlighted with a blue background.

DISK
0x0001
0x0002
0x0007
0x0008
...
0x00F5

2nd Level Page Table

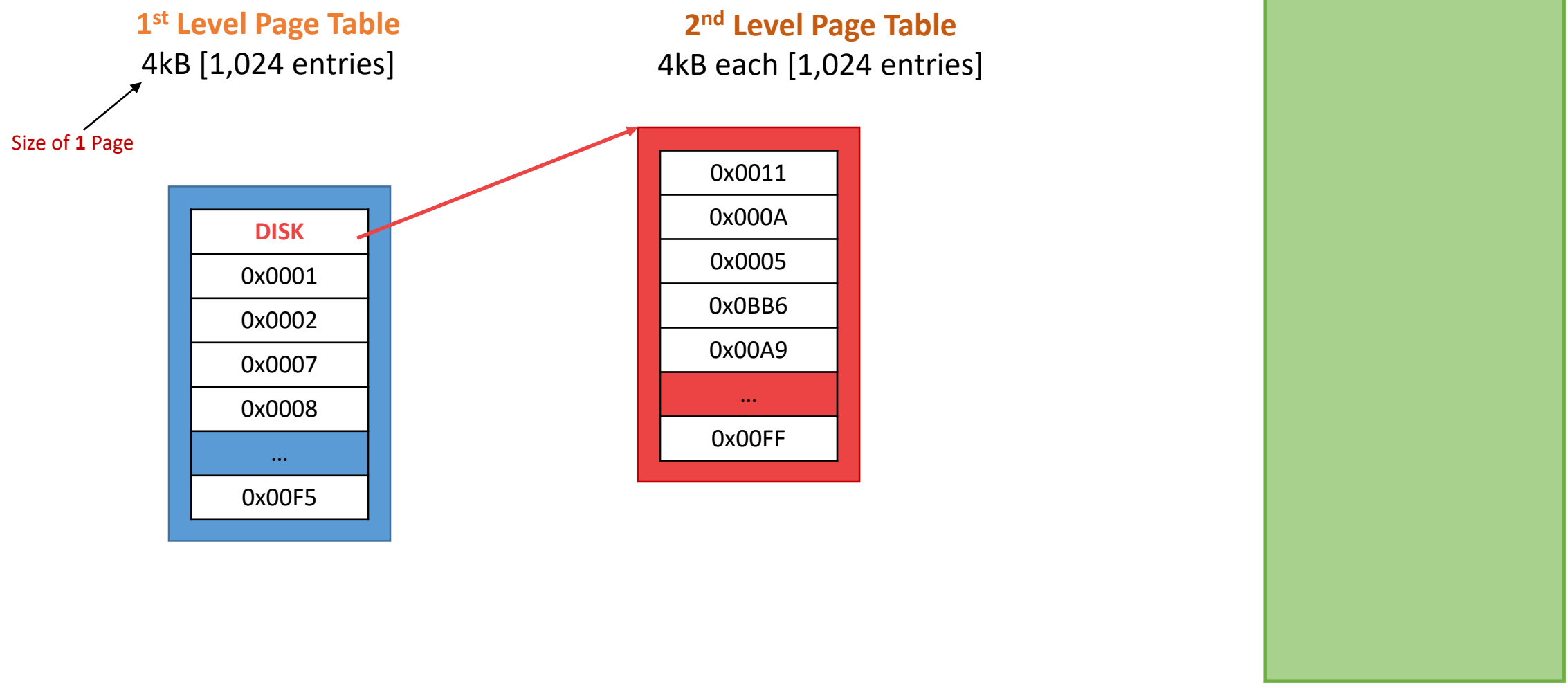
4kB each [1,024 entries]

Main Memory



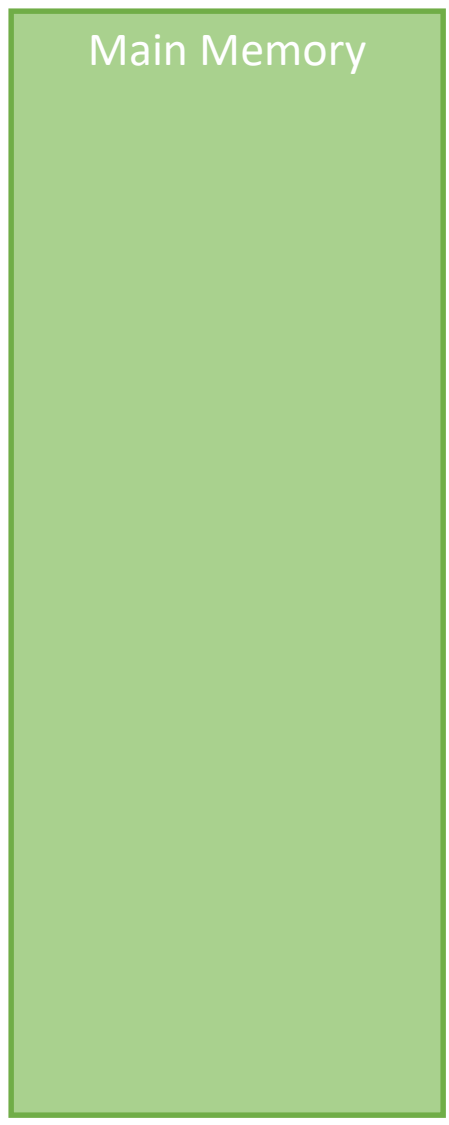
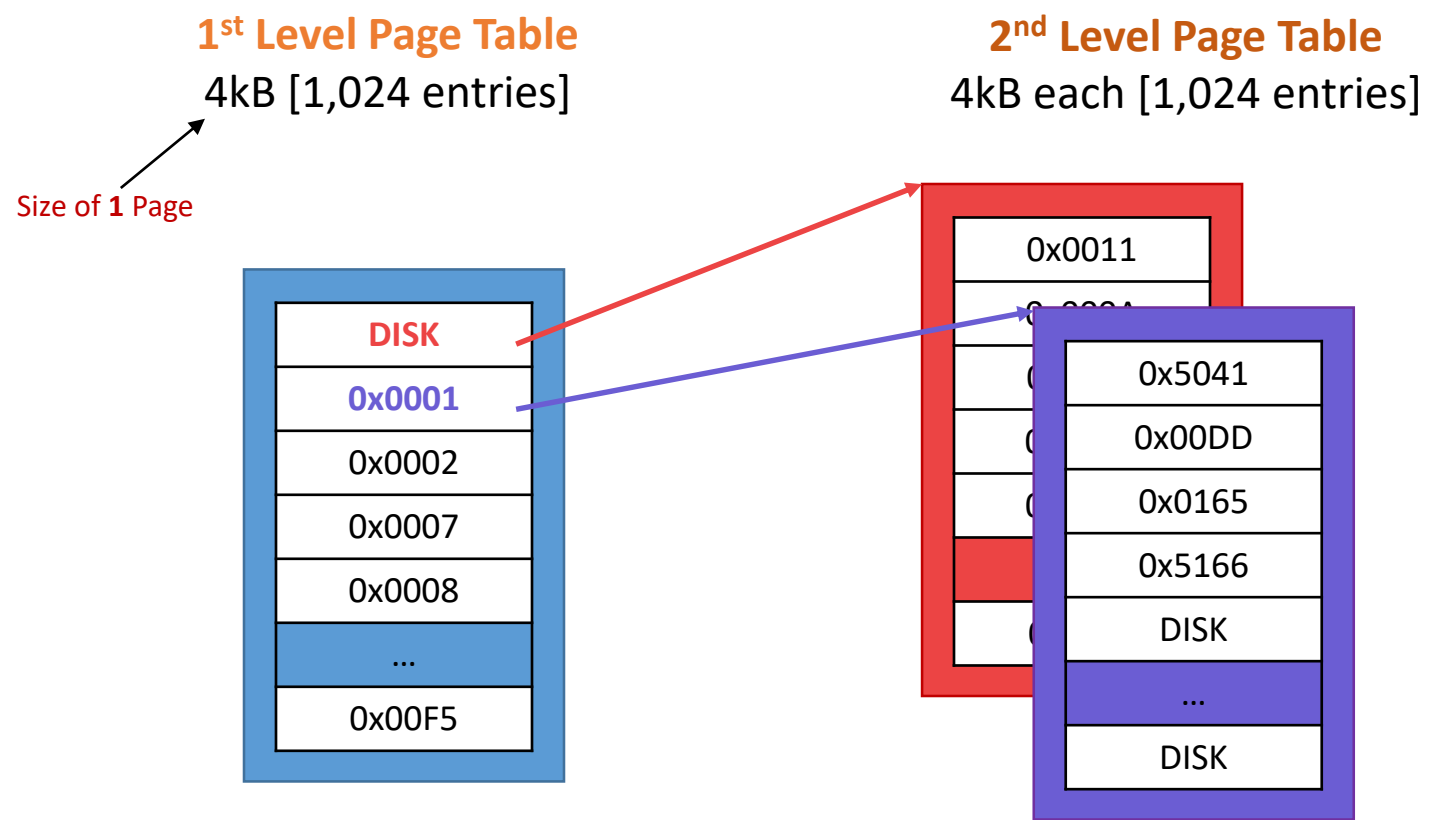
Quincy Flint

Multi-Level Page Tables



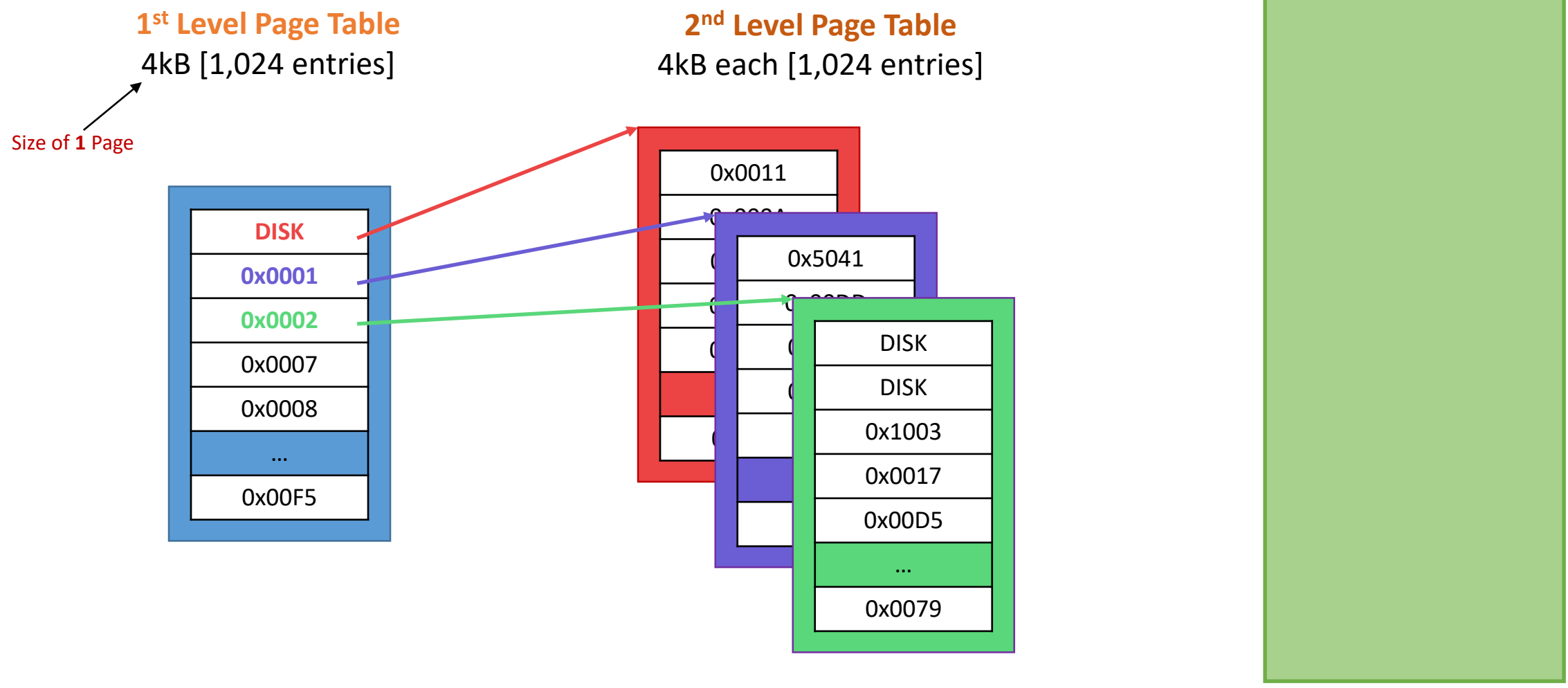
Quincy Flint

Multi-Level Page Tables



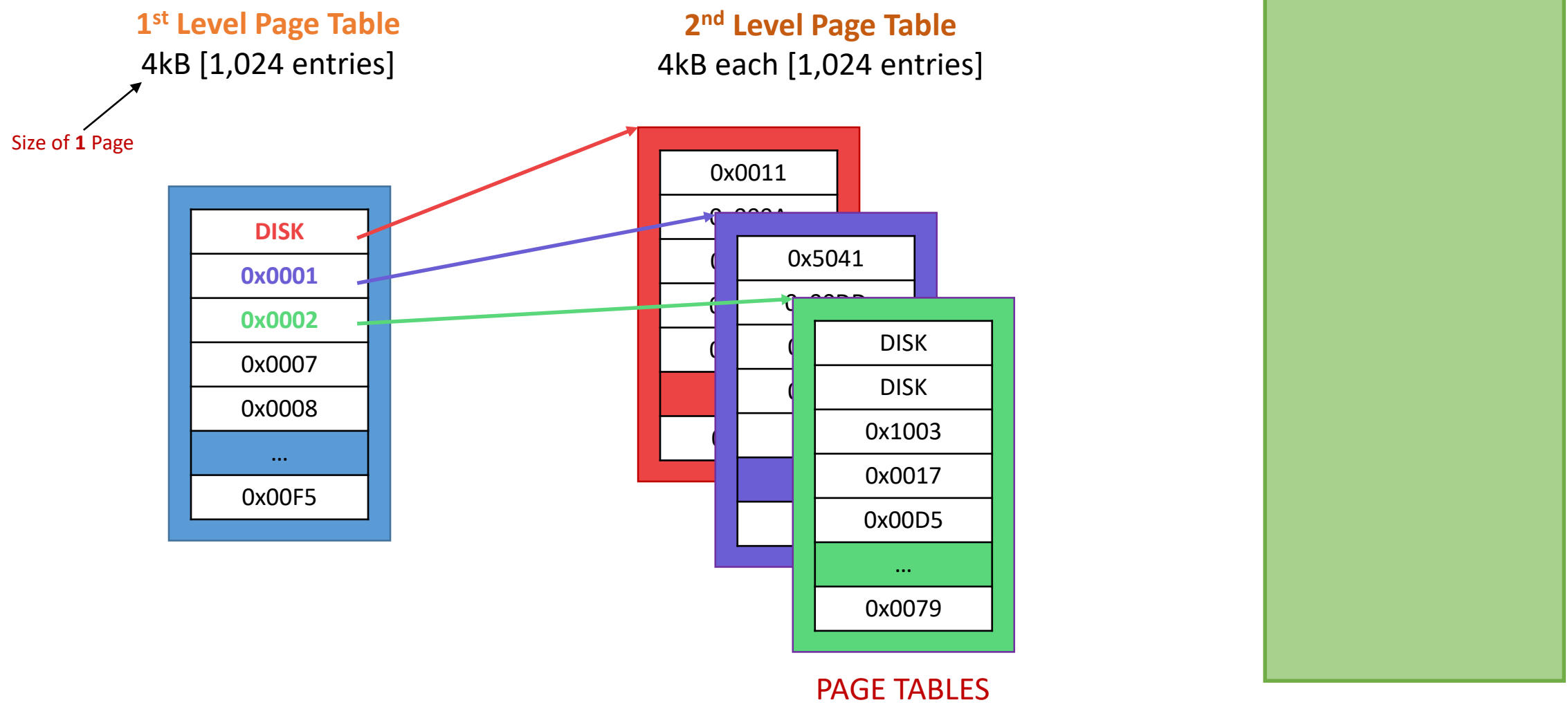
Quincy Flint

Multi-Level Page Tables



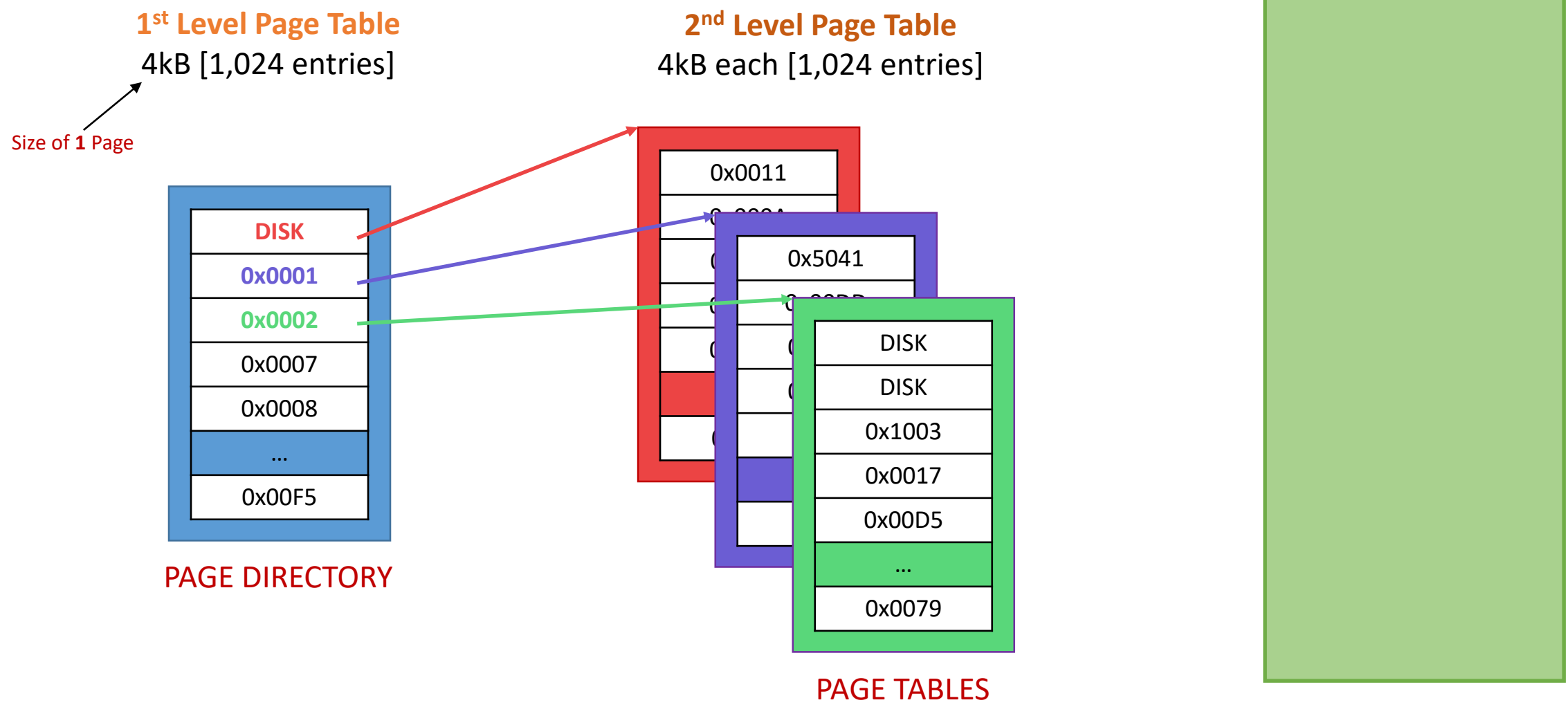
Quincy Flint

Multi-Level Page Tables



Quincy Flint

Multi-Level Page Tables



Quincy Flint

Multi-Level Page Tables

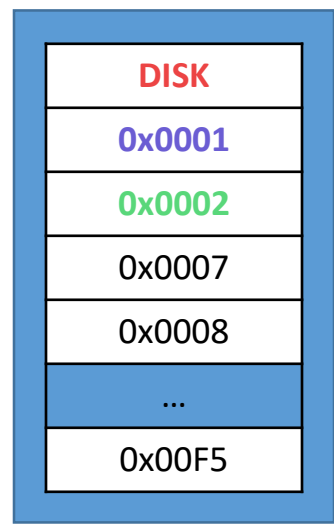
1st Level Page Table

4kB [1,024 entries]

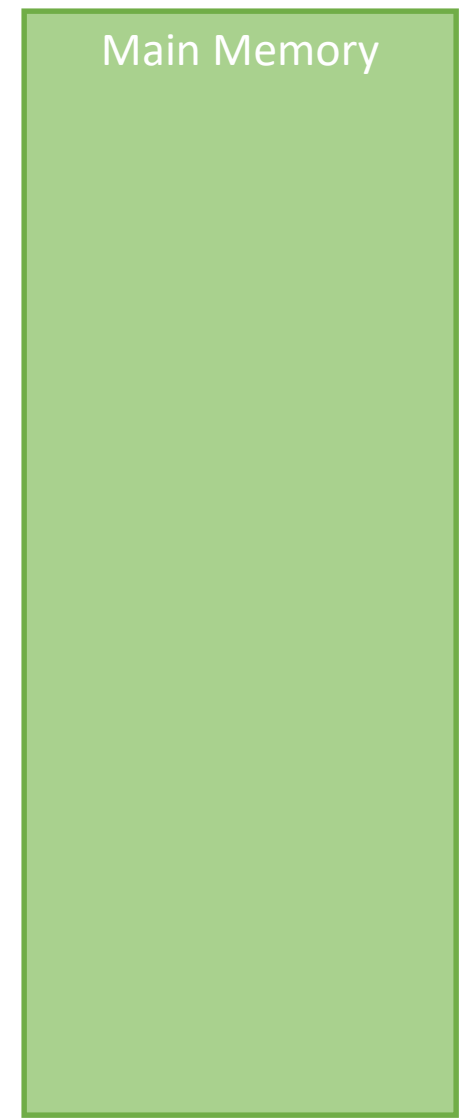
Size of 1 Page

2nd Level Page Table

4kB each [1,024 entries]

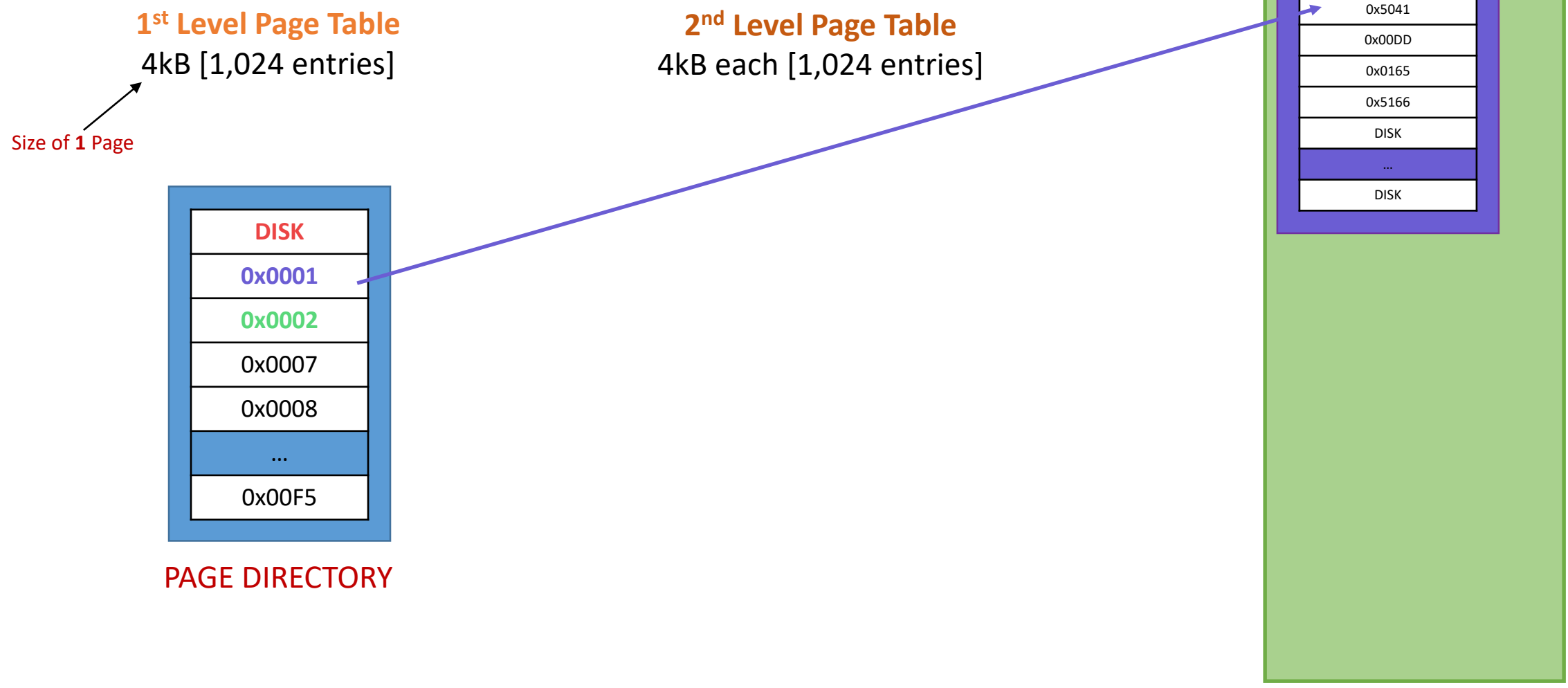


PAGE DIRECTORY



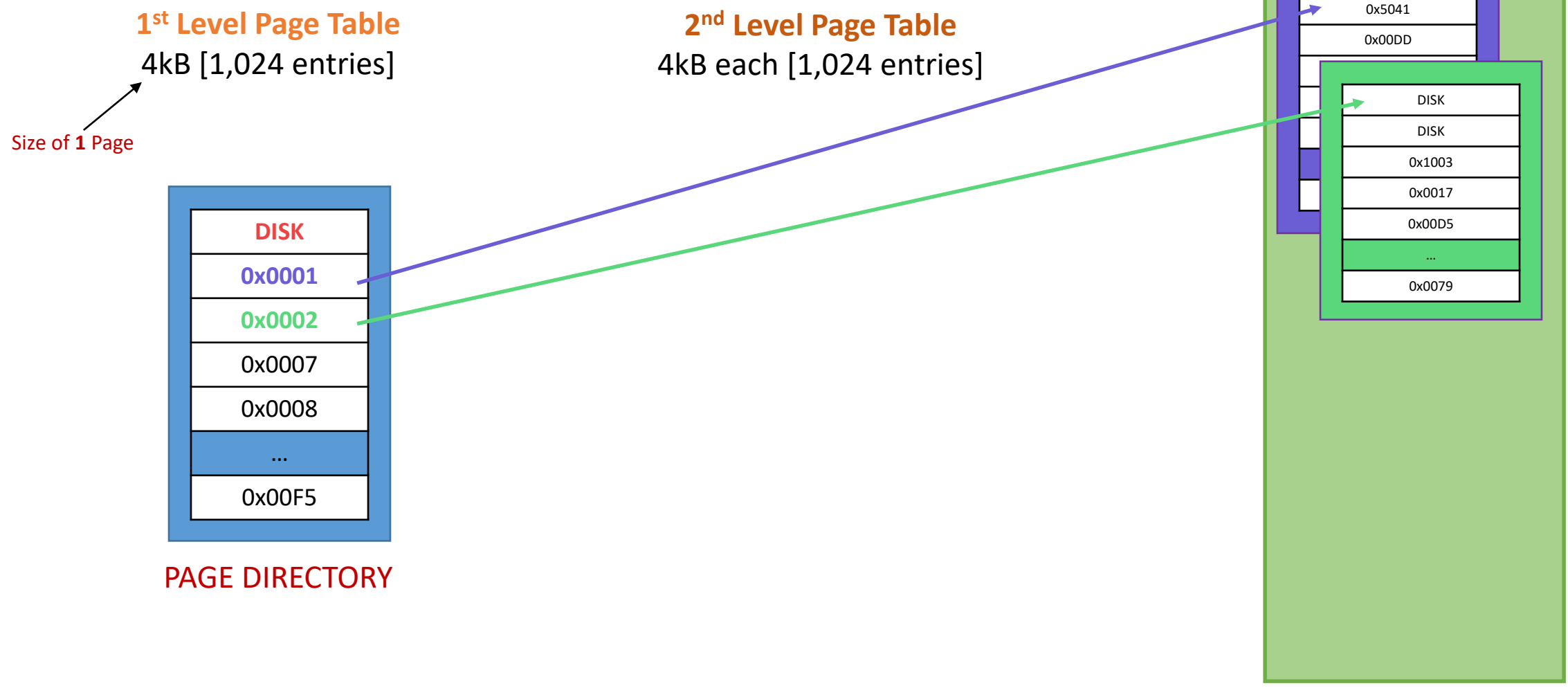
Quincy Flint

Multi-Level Page Tables



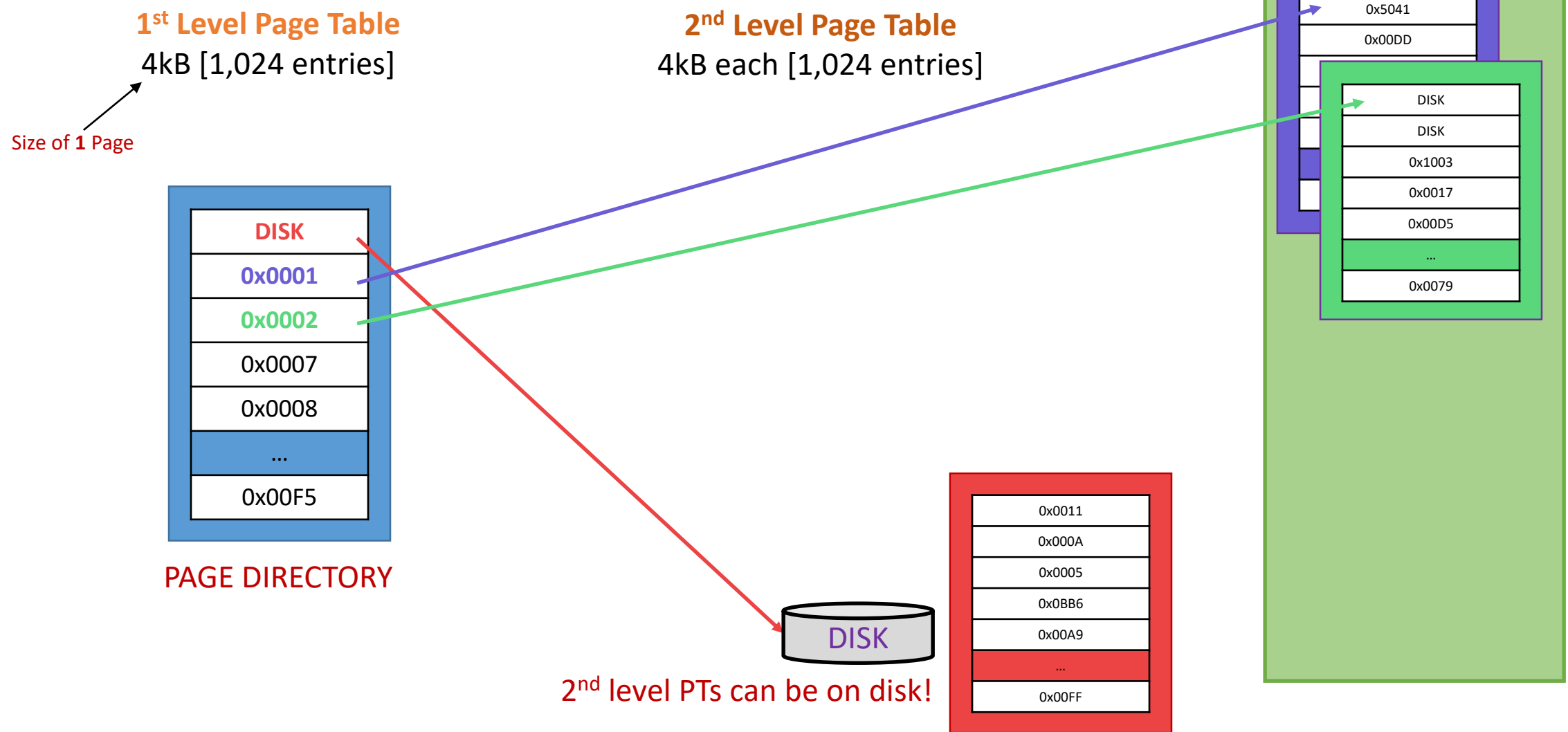
Quincy Flint

Multi-Level Page Tables



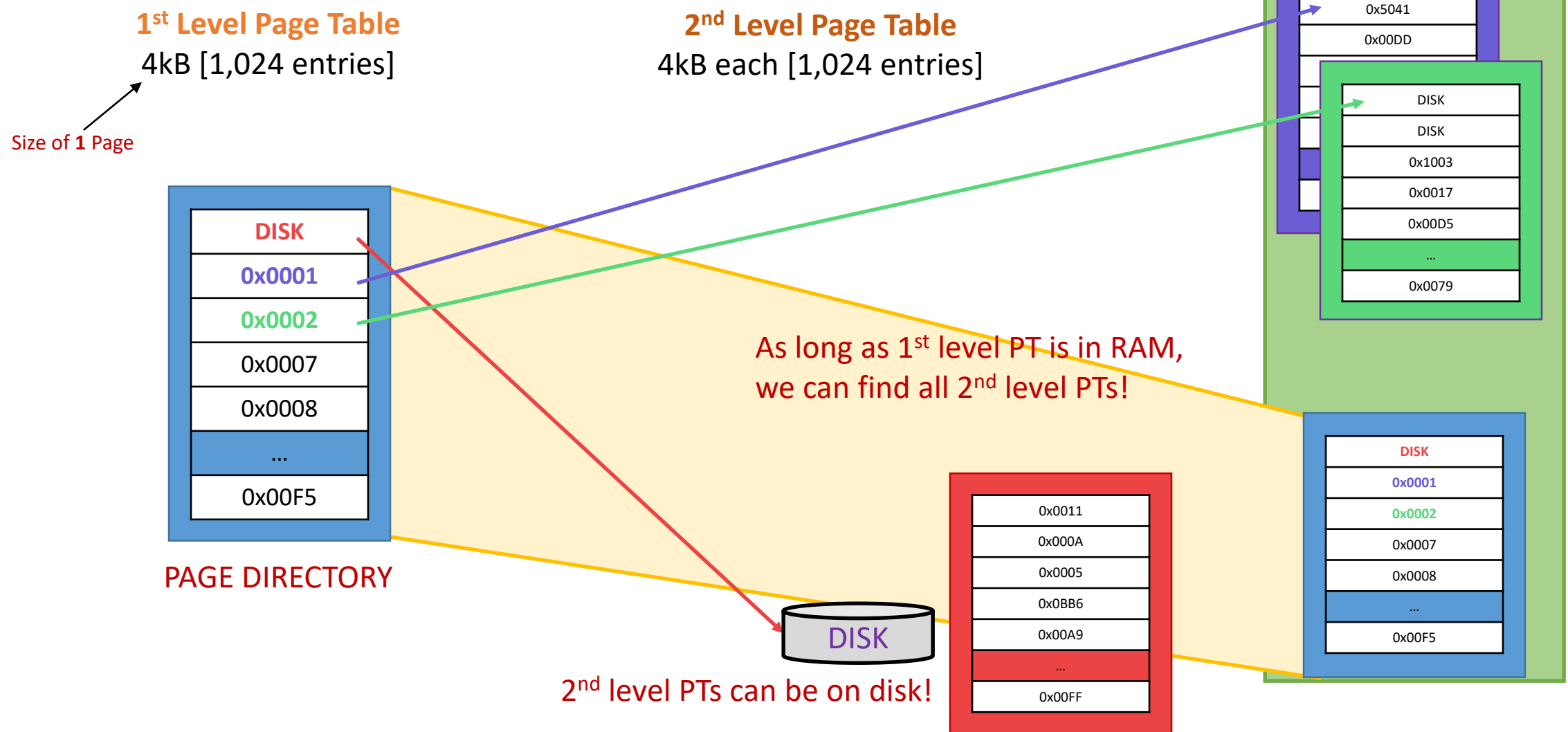
Quincy Flint

Multi-Level Page Tables



Quincy Flint

Multi-Level Page Tables



Quincy Flint

Quiz: Multi-Level Page Tables

Q: With multilevel page tables, what is the smallest amount of page table data that we need in RAM to run a single 32-bit application?

- I. 4 kB
- II. 8 kB
- III. 8 MB
- IV. 1 GB
- V. 4 MB

Quincy Flint

Quiz: Multi-Level Page Tables

Q: With multilevel page tables, what is the smallest amount of page table data that we need in RAM to run a single 32-bit application?

- I. 4 kB
- II. 8 kB
- III. 8 MB
- IV. 1 GB
- V. 4 MB

A:

- II. 8 kB

We must always keep the 1st level page table in RAM (4 kB) and we need at least the 2nd level page table for the application data.

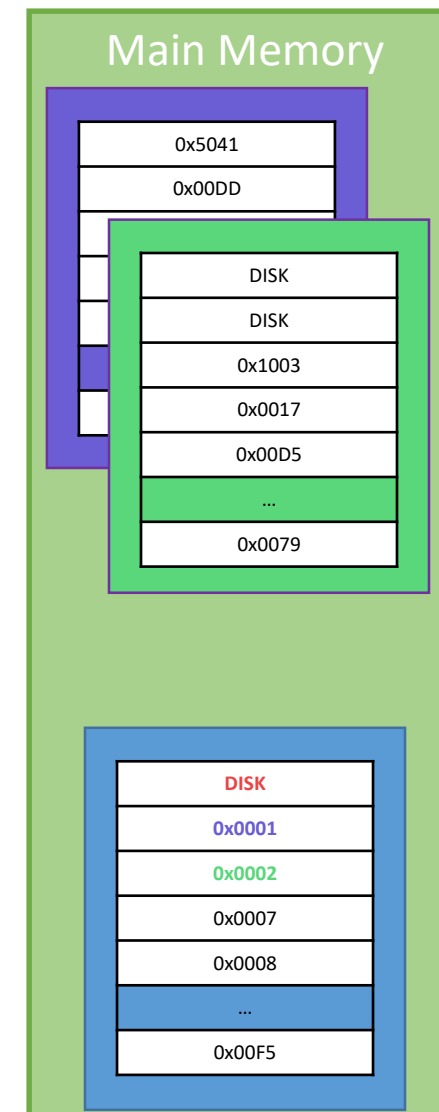
Quincy Flint

Multi-Level Page Table Translation

Virtual Address [32 bit]



Physical Address [28 bits]



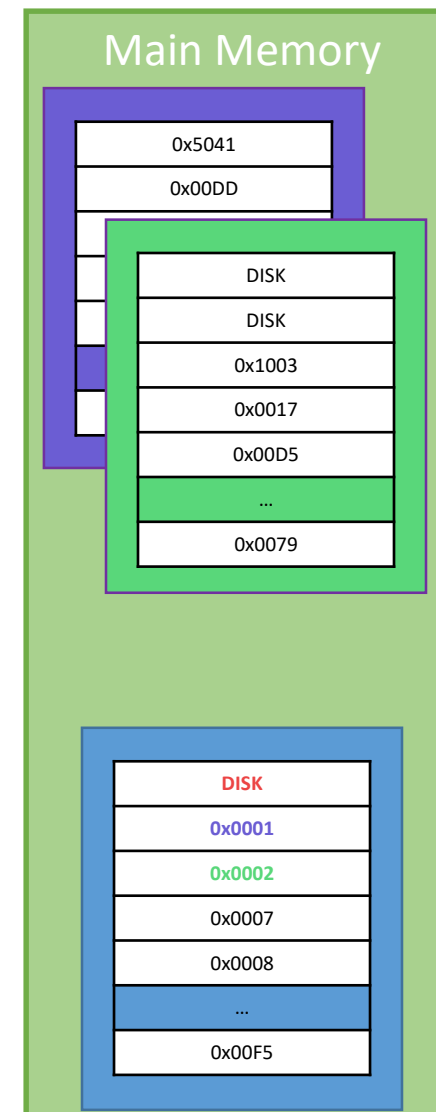
Quincy Flint

Multi-Level Page Table Translation

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

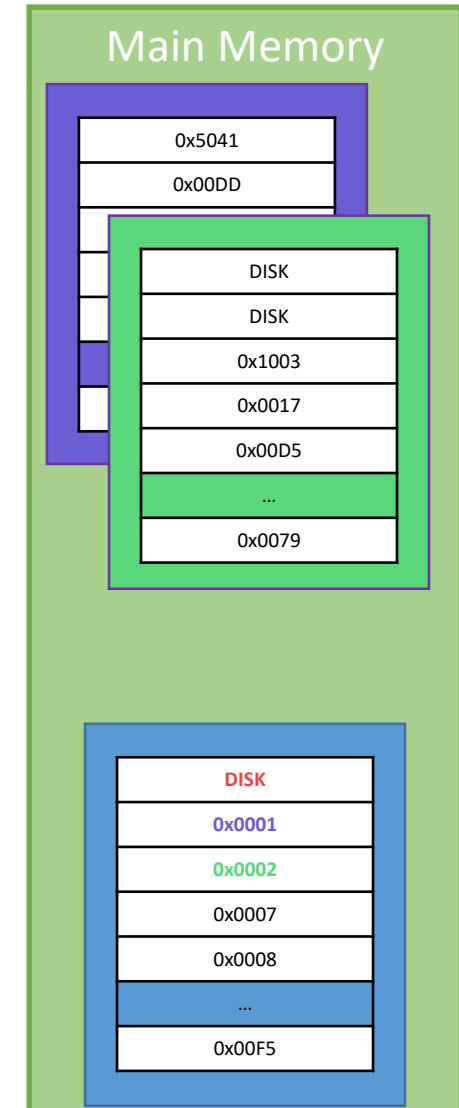
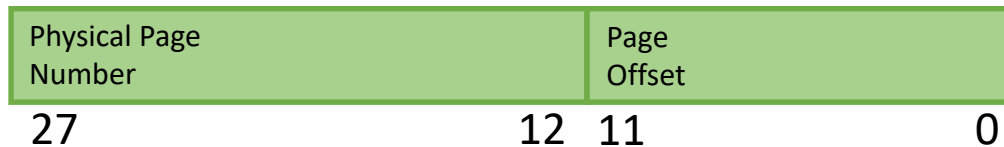
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

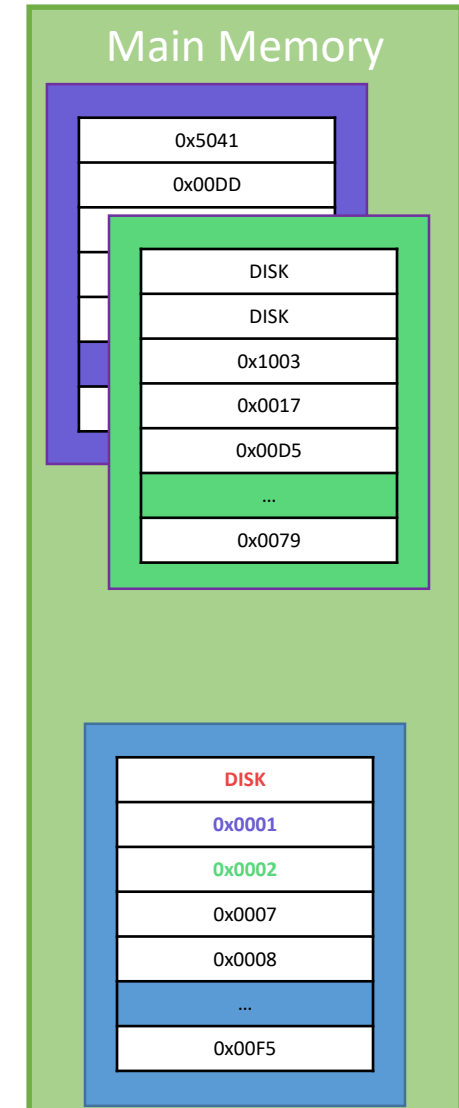
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

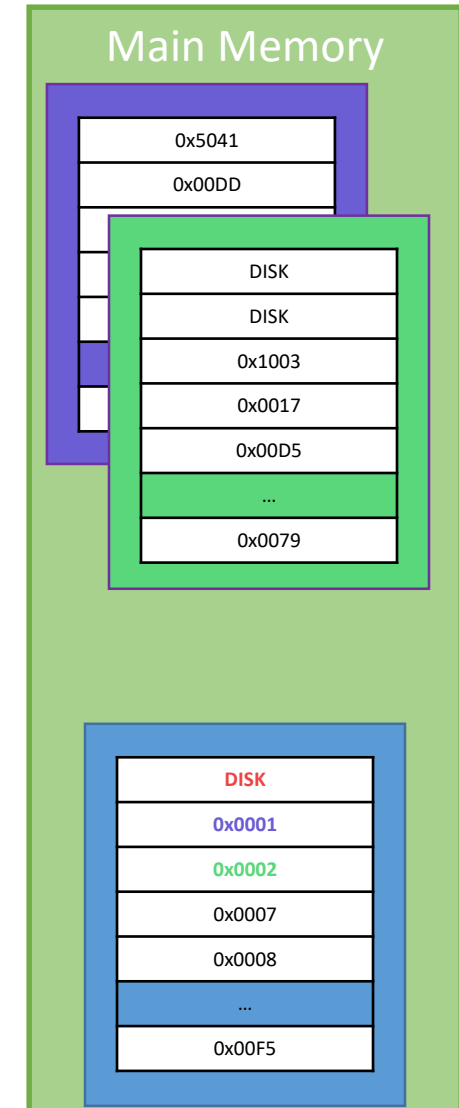
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

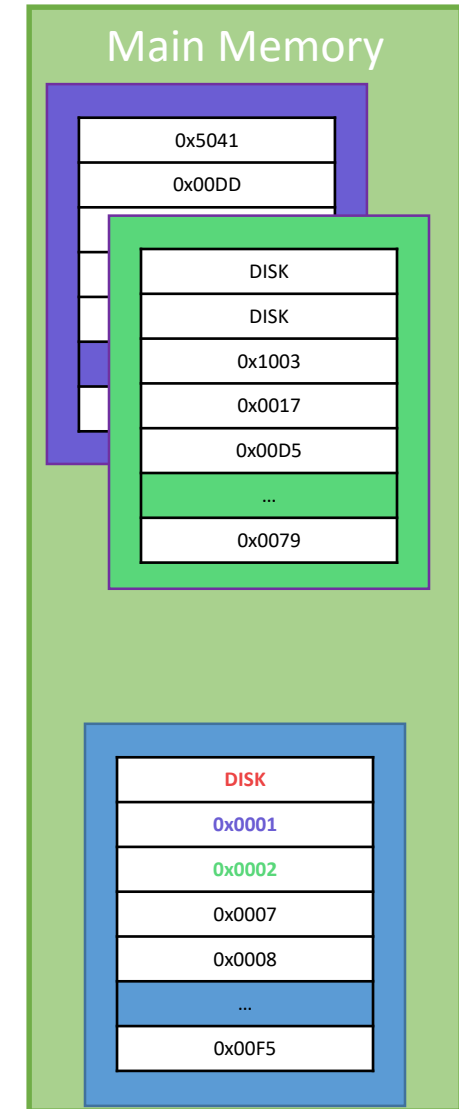
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

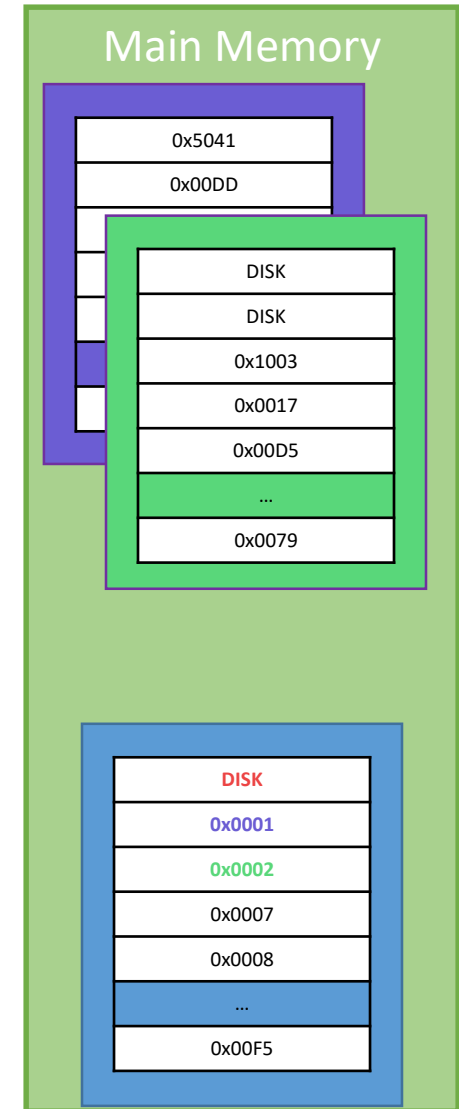
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

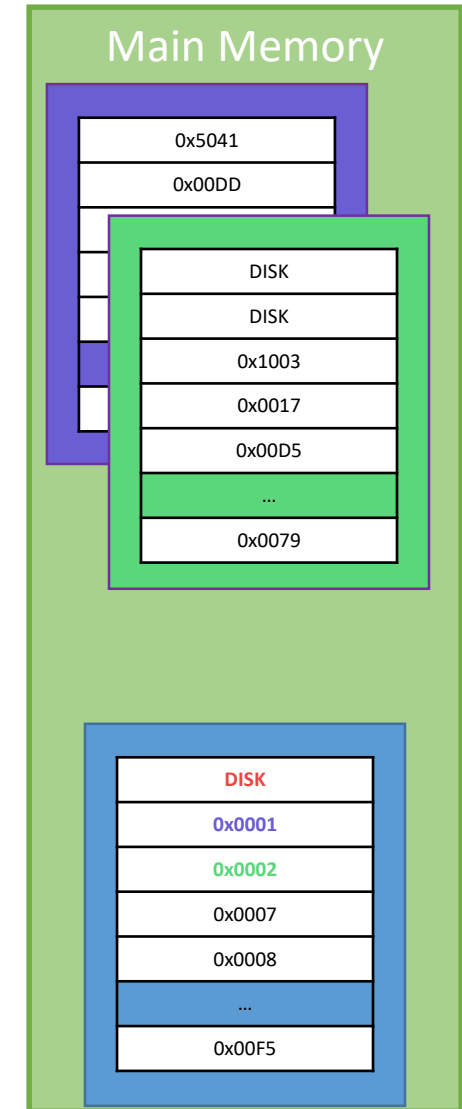
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]



Quincy Flint

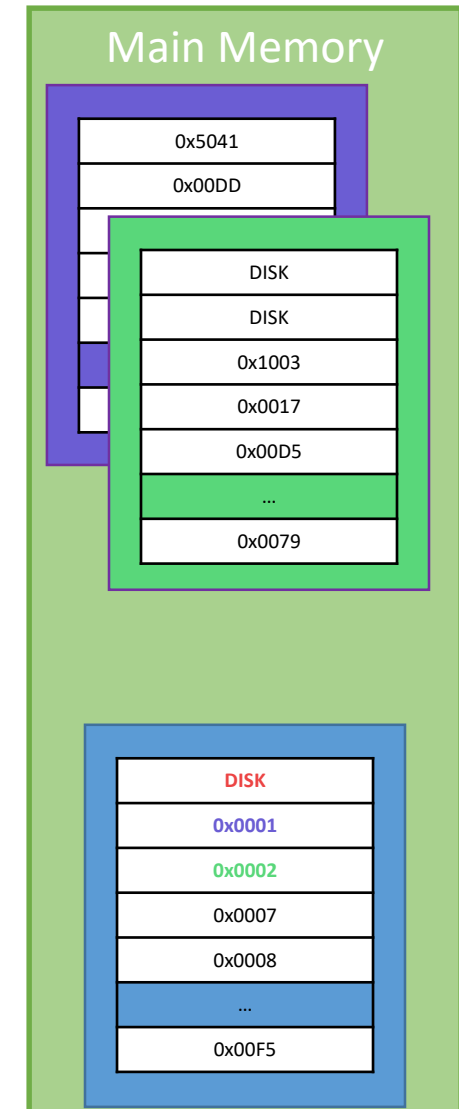
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

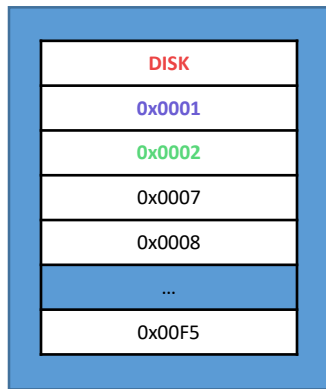


Quincy Flint

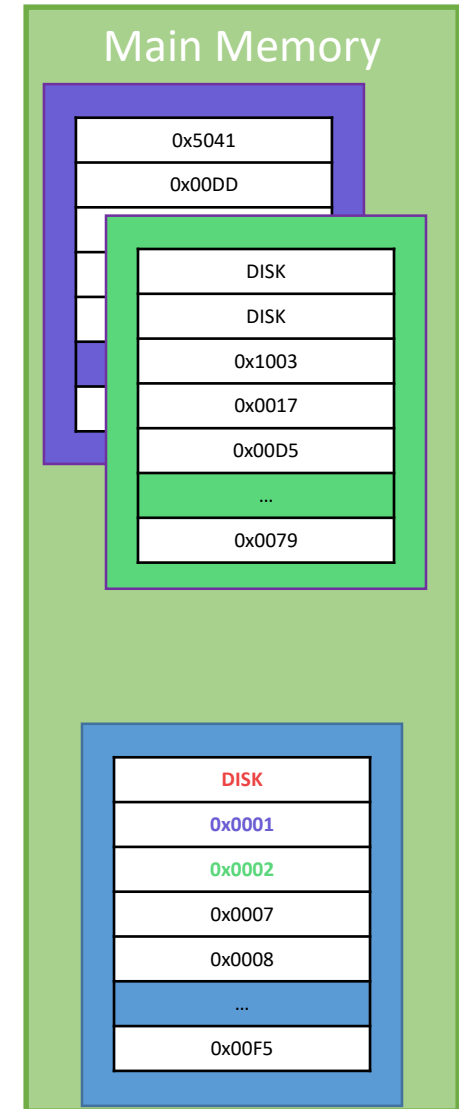
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

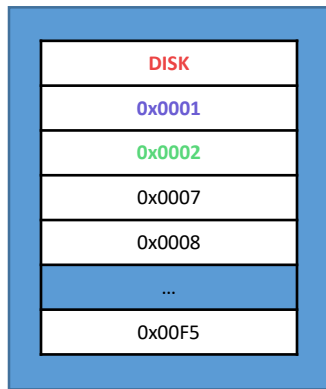


Quincy Flint

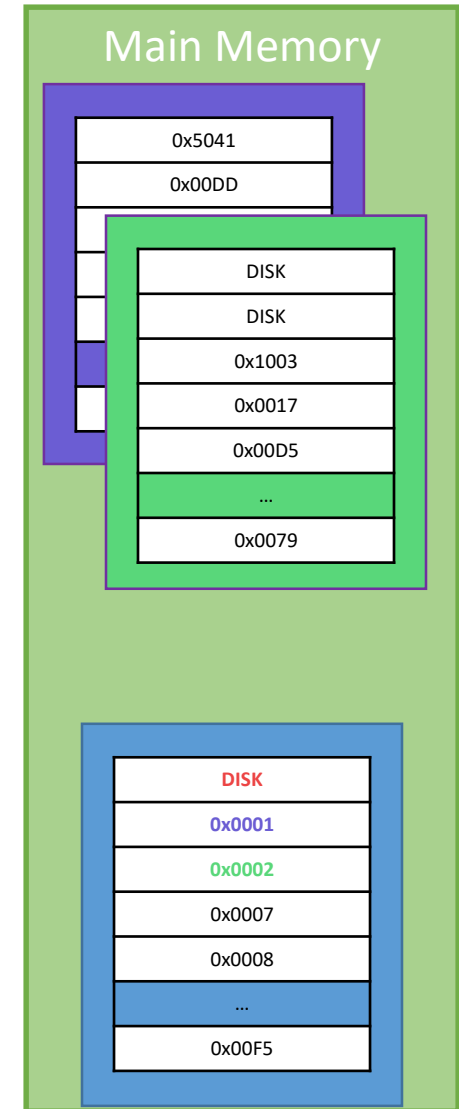
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

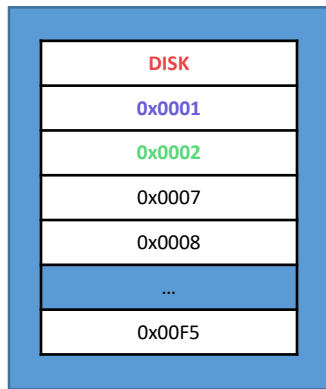


Quincy Flint

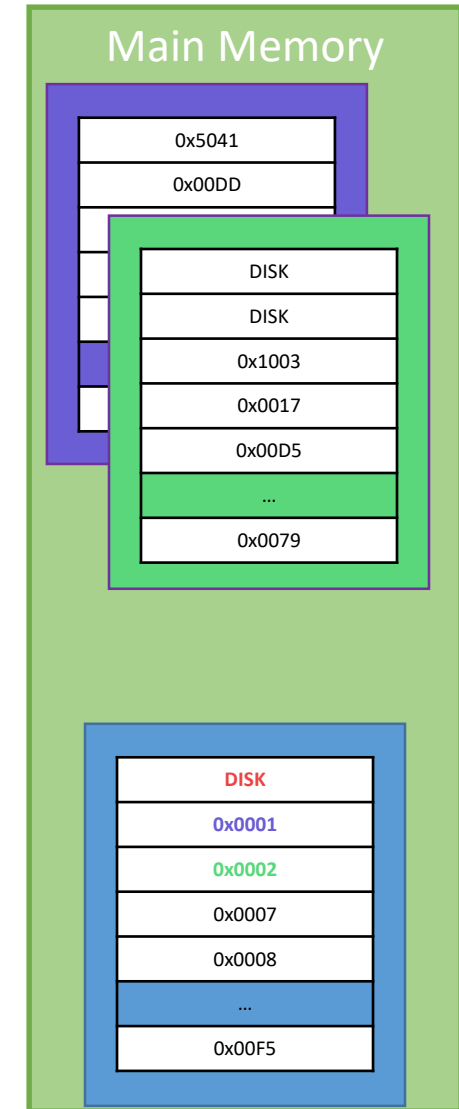
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

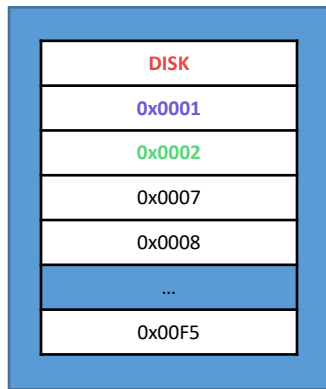


Quincy Flint

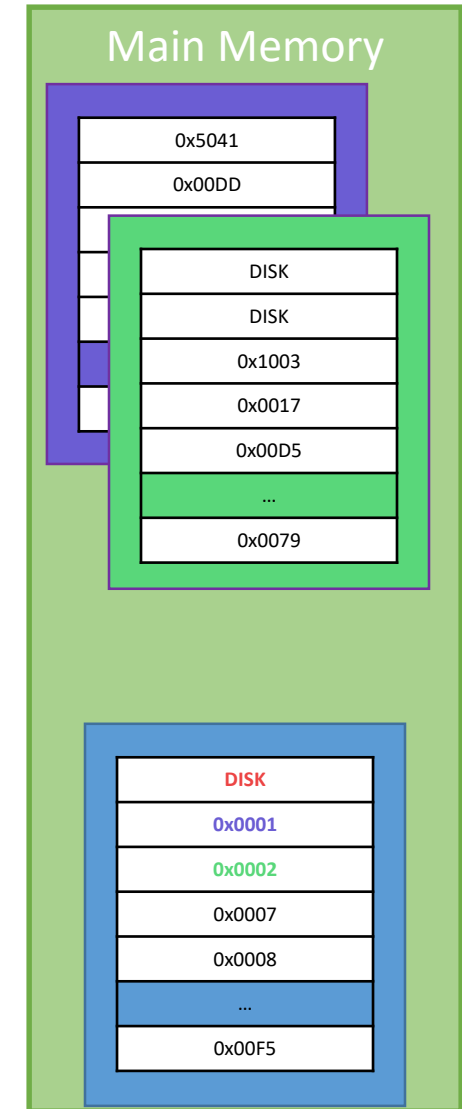
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

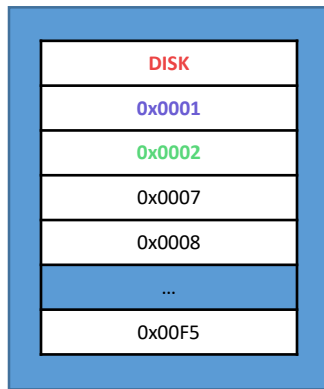


Quincy Flint

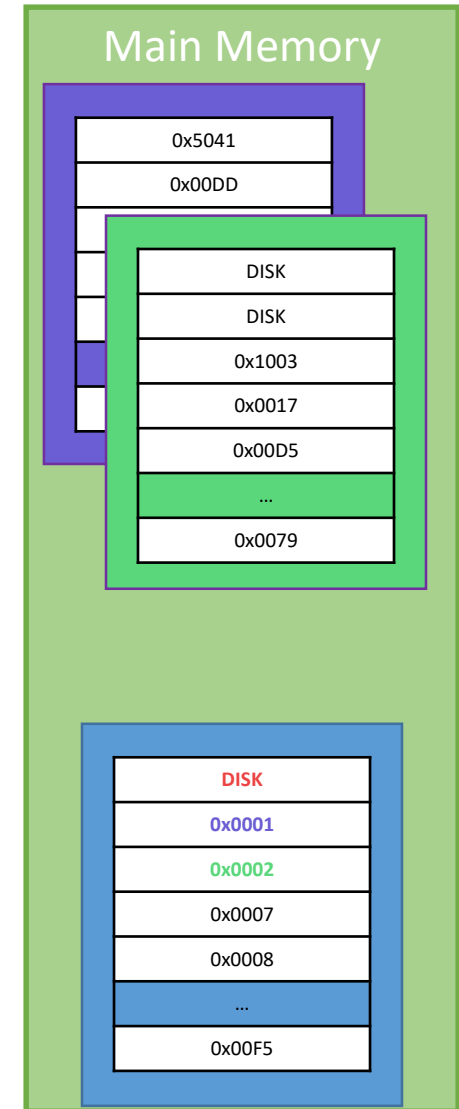
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

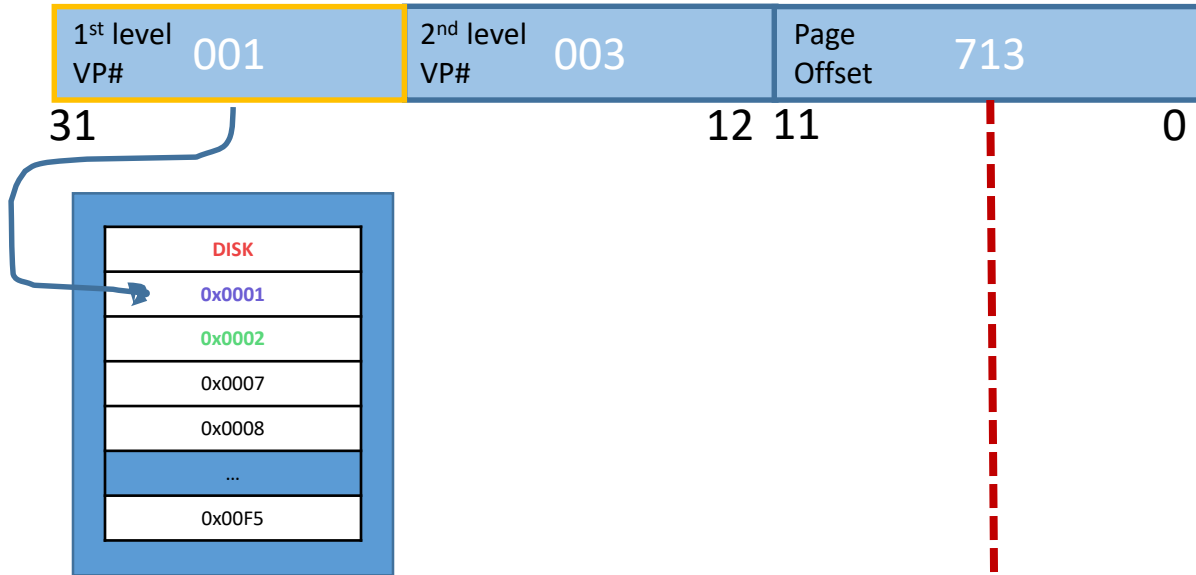


Quincy Flint

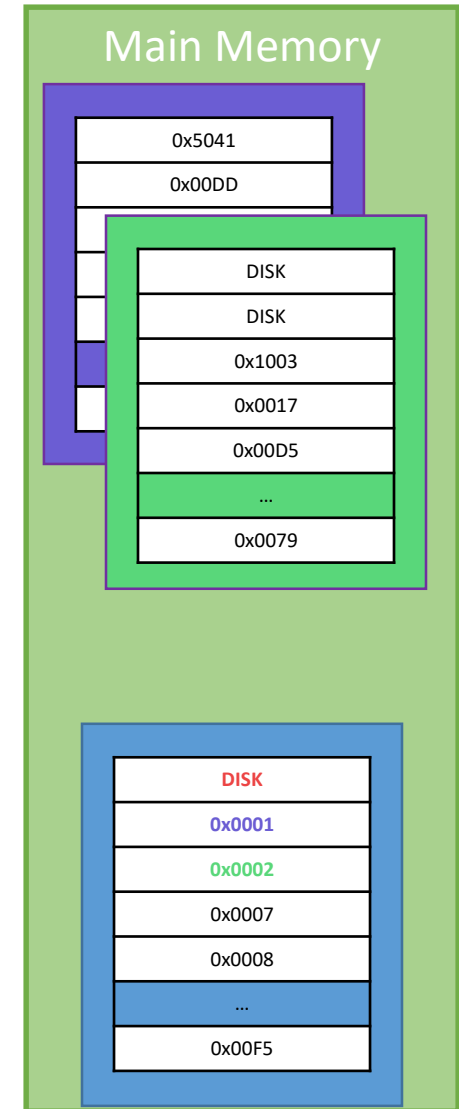
Multi-Level Page Table Translation

VA: **0x00403 713**

Virtual Address [32 bit]



Physical Address [28 bits]

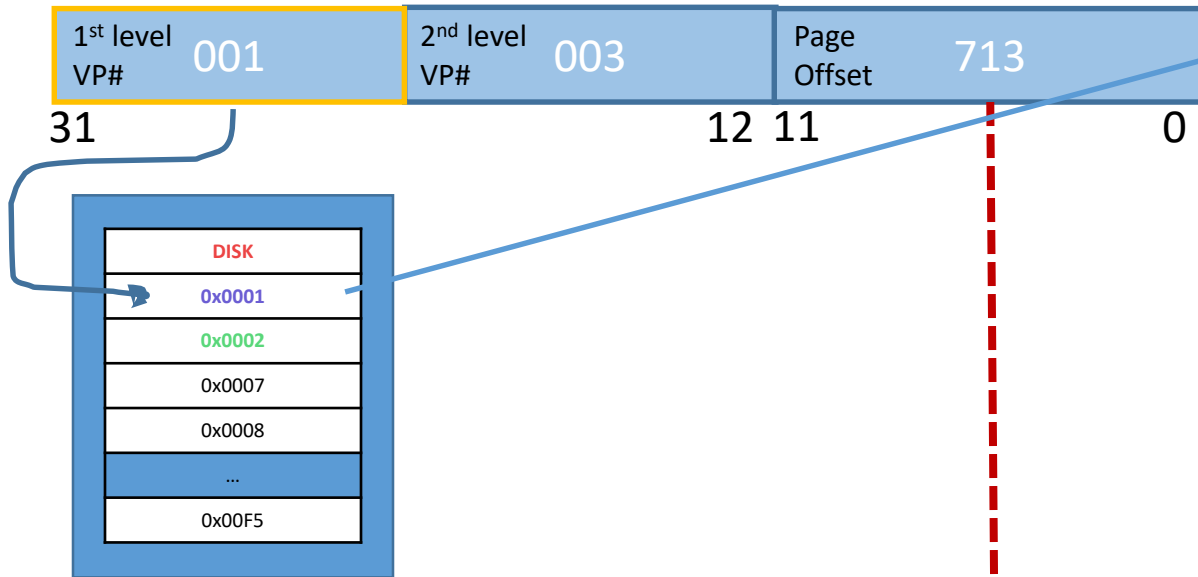


Quincy Flint

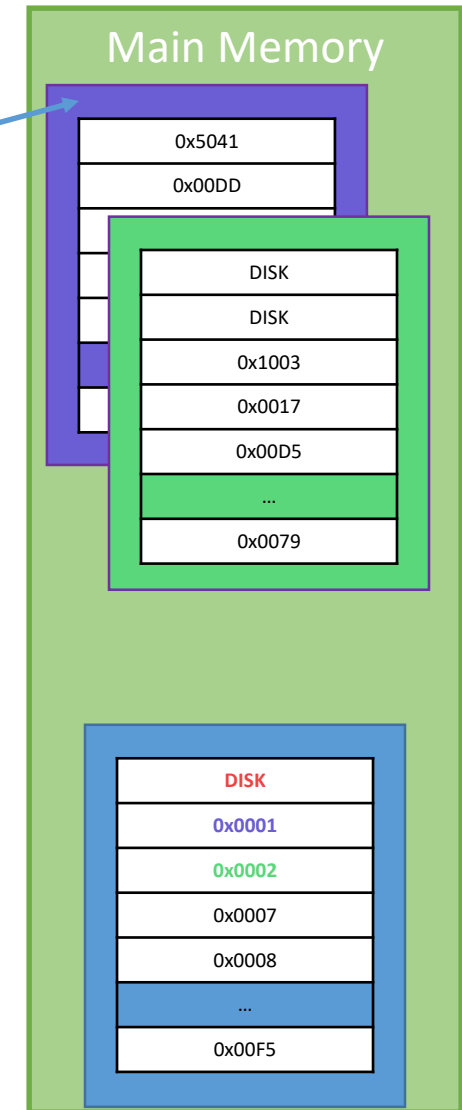
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

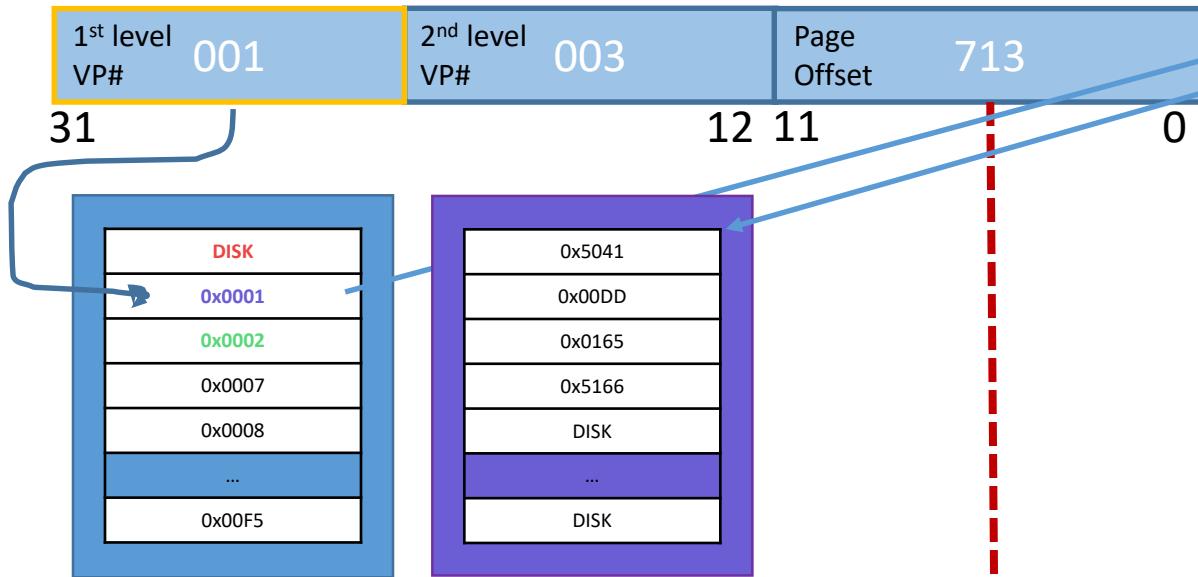


Quincy Flint

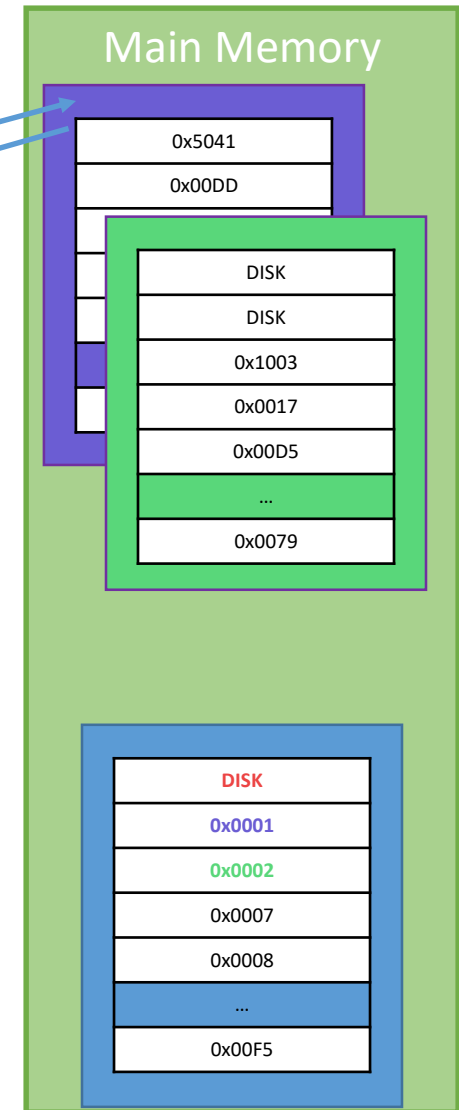
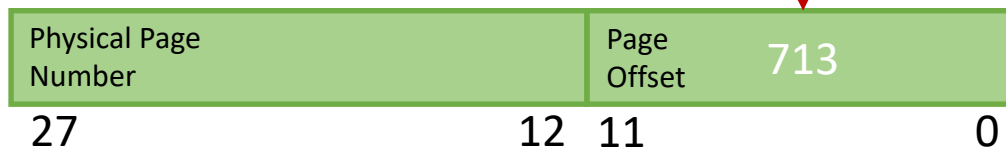
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

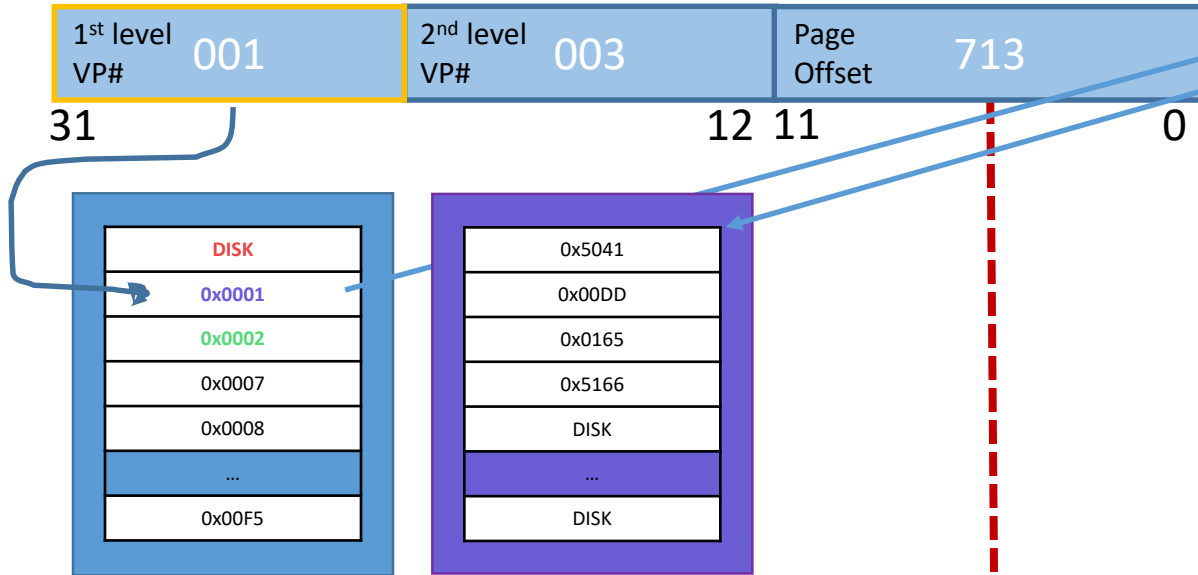


Quincy Flint

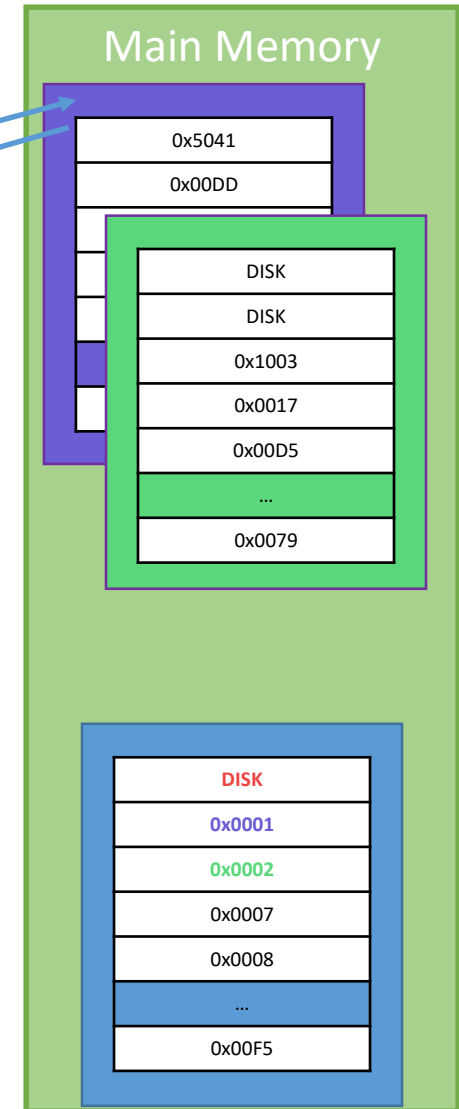
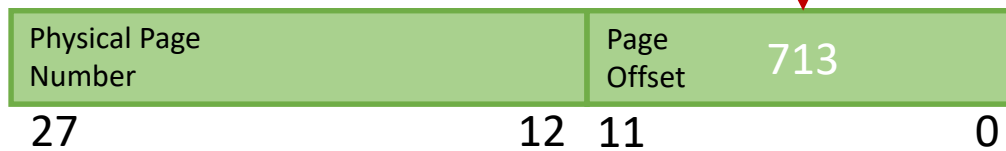
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

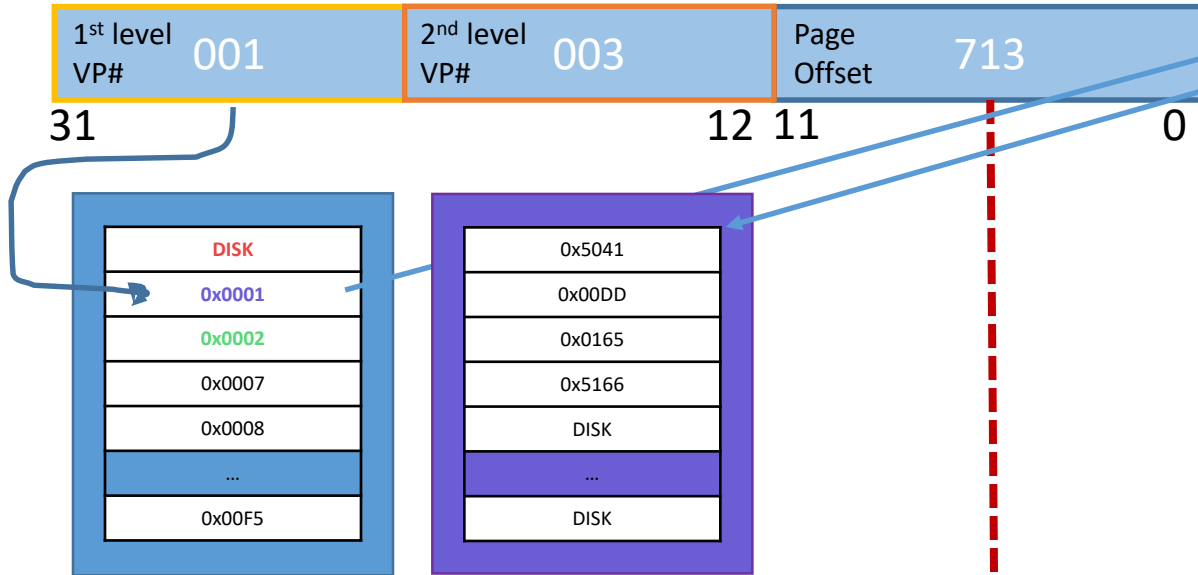


Quincy Flint

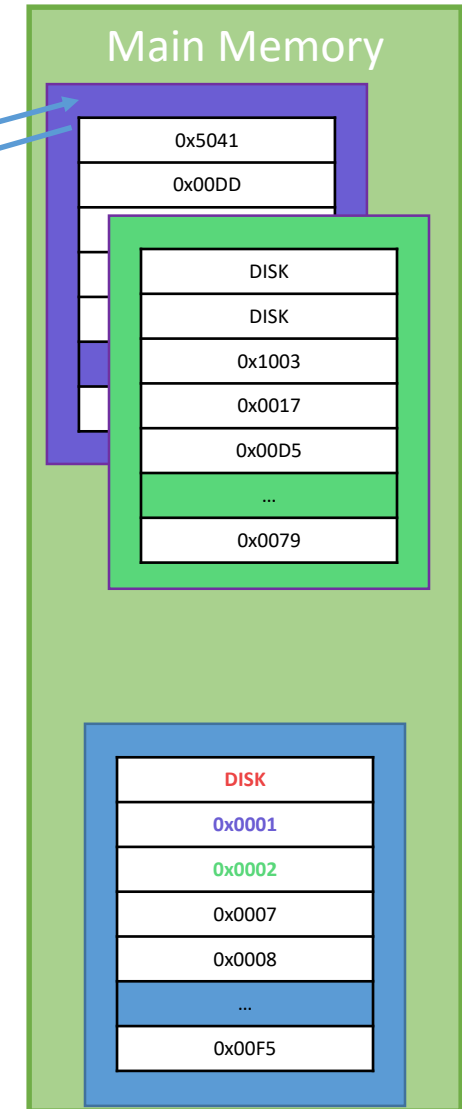
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

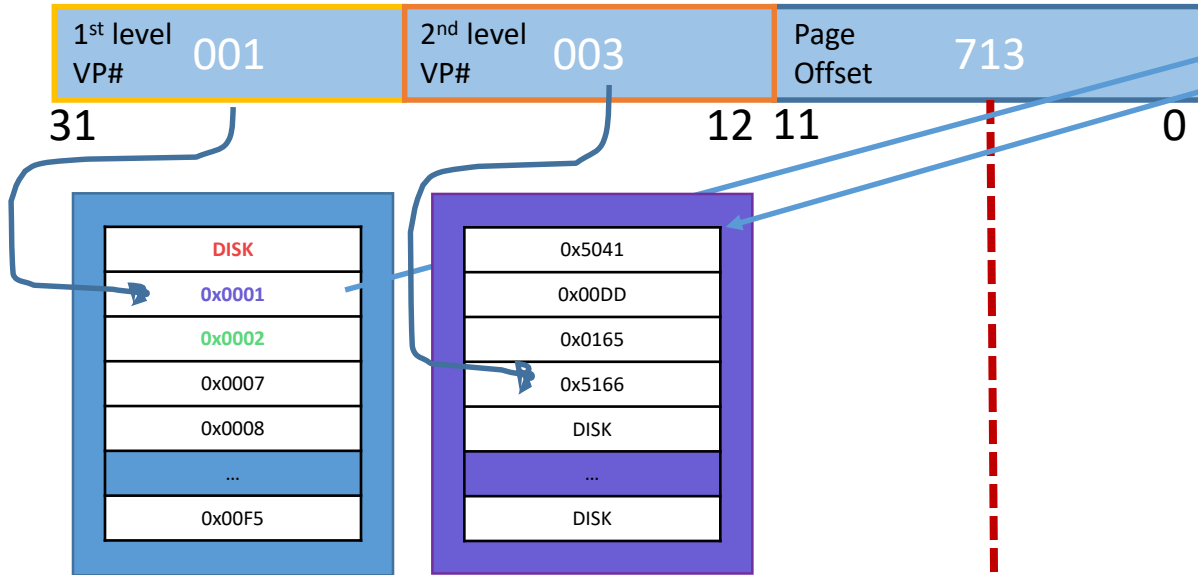


Quincy Flint

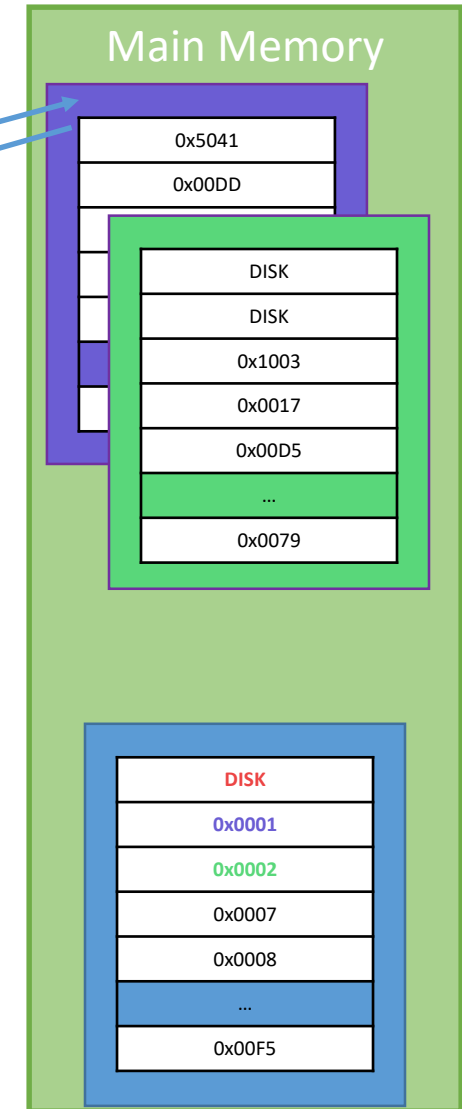
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

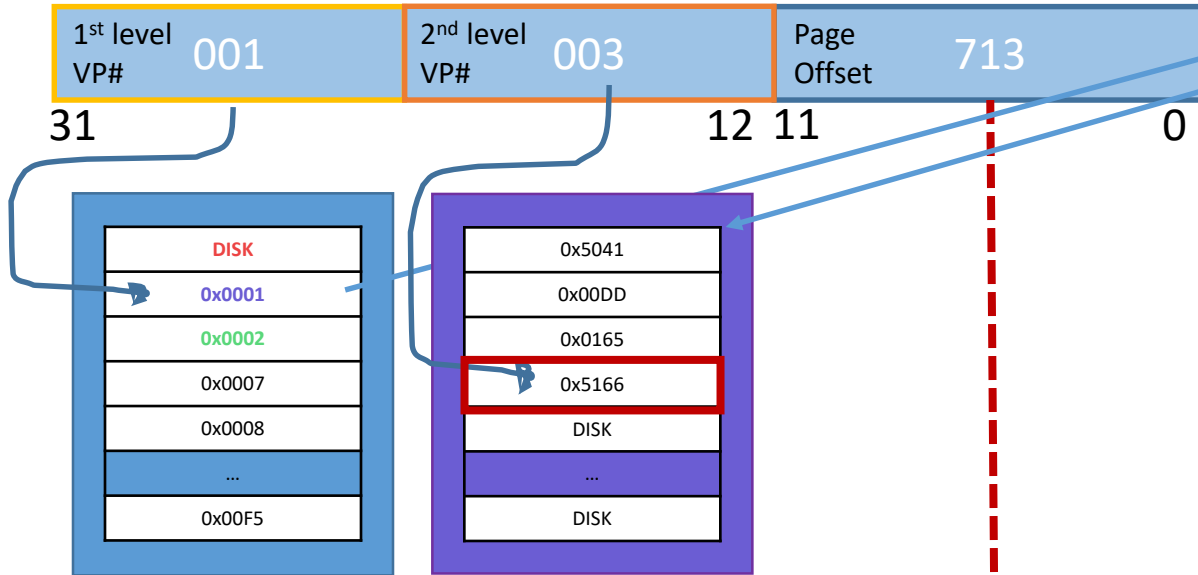


Quincy Flint

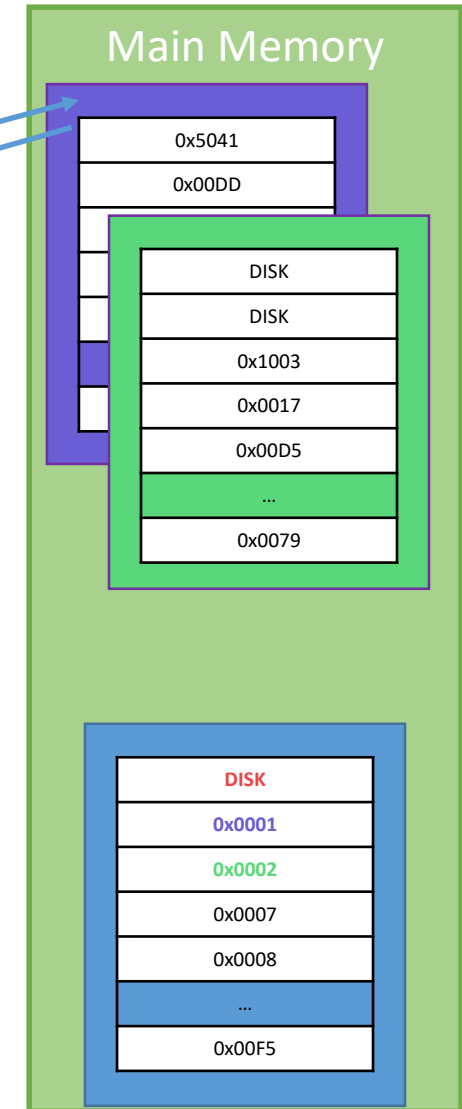
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

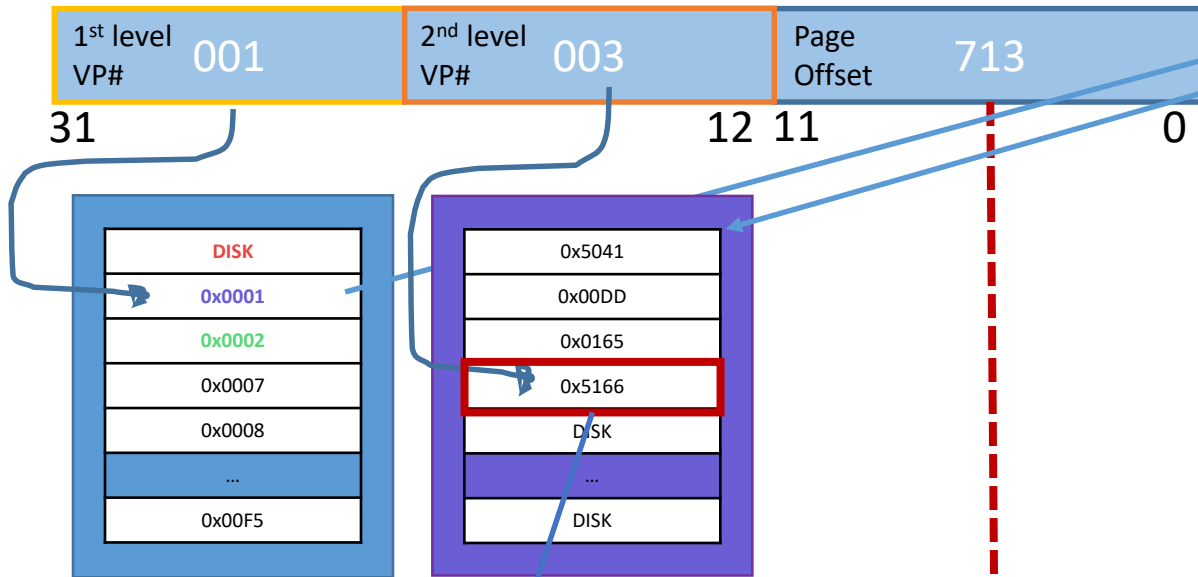


Quincy Flint

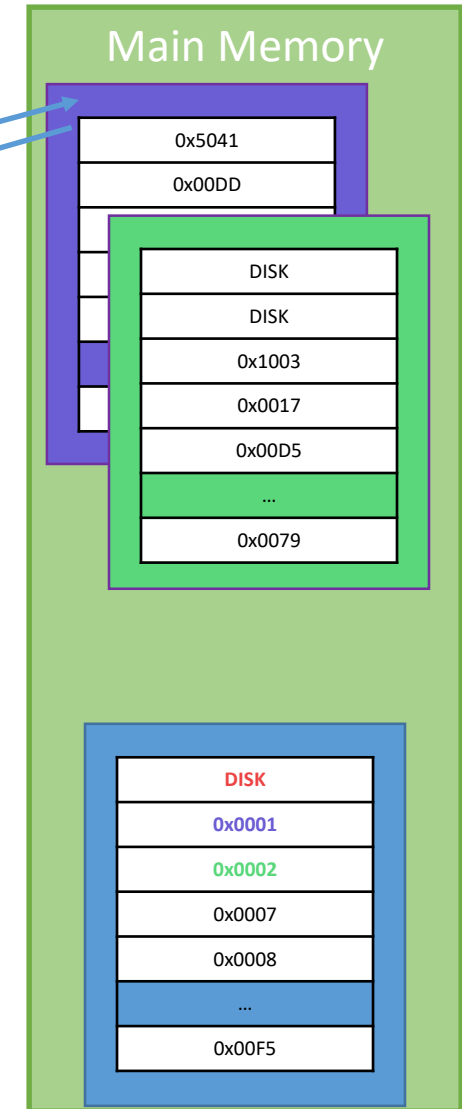
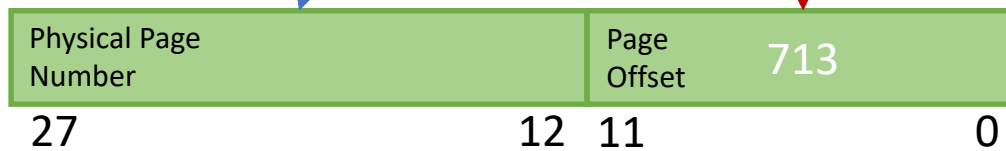
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

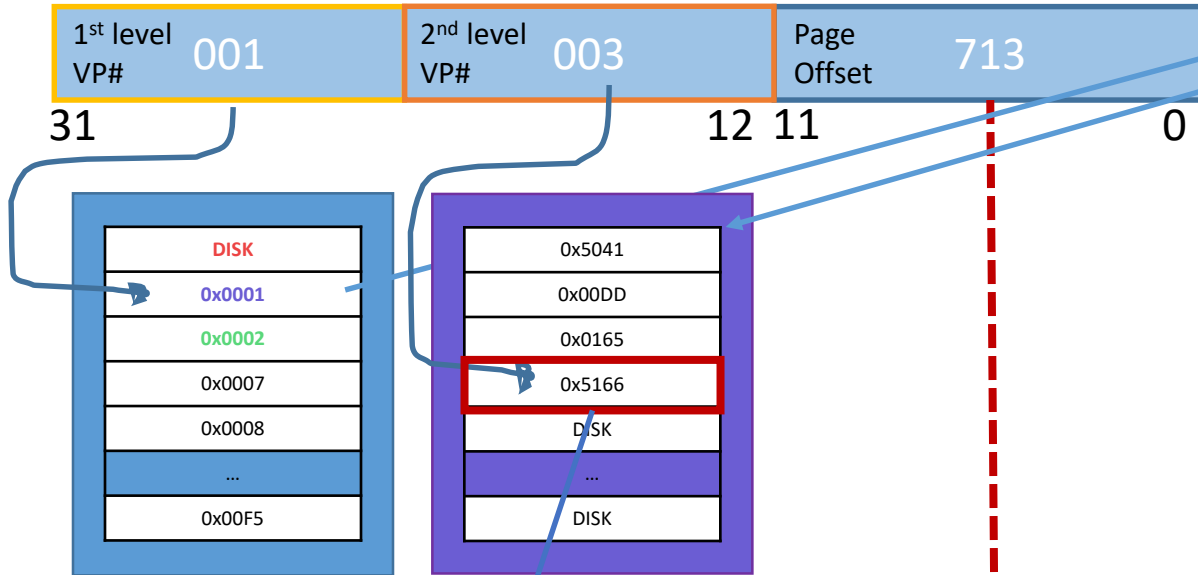


Quincy Flint

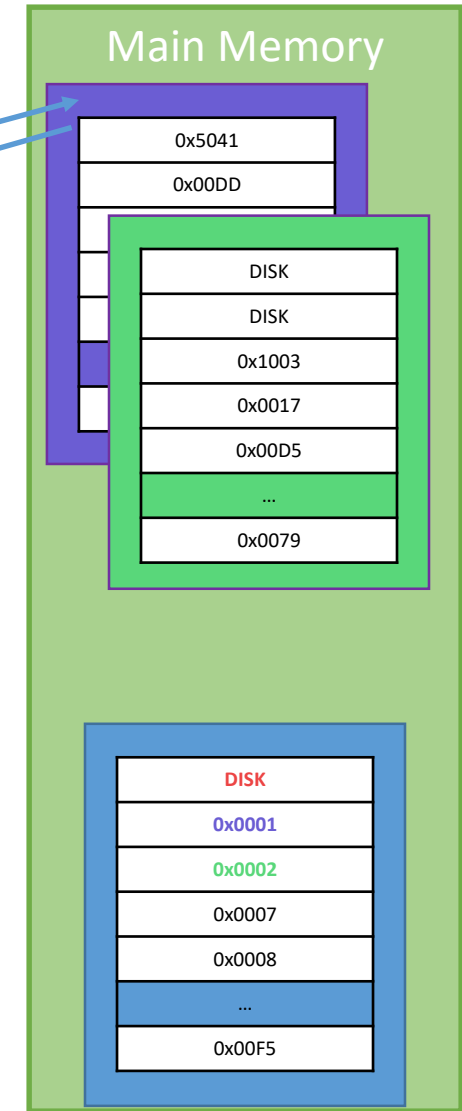
Multi-Level Page Table Translation

VA: 0x00403 713

Virtual Address [32 bit]



Physical Address [28 bits]

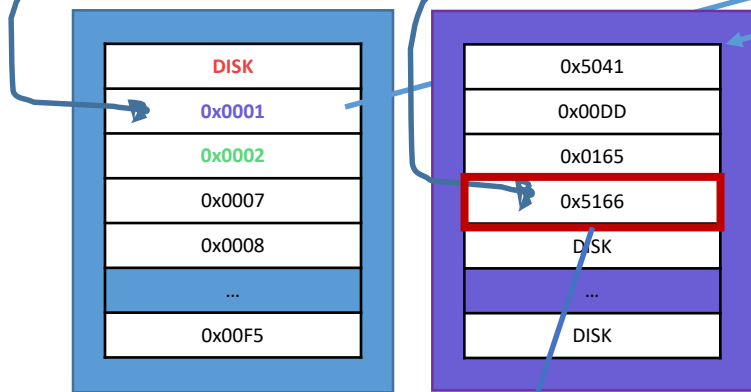
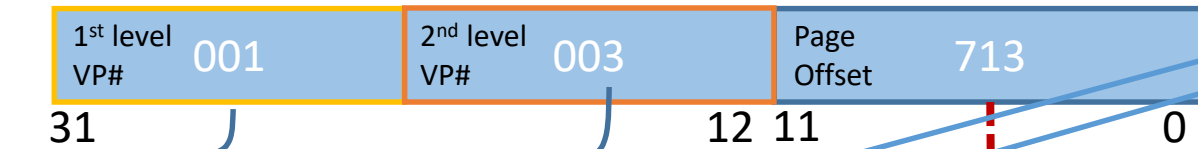


Quincy Flint

Multi-Level Page Table Translation

VA: 0x00403 713

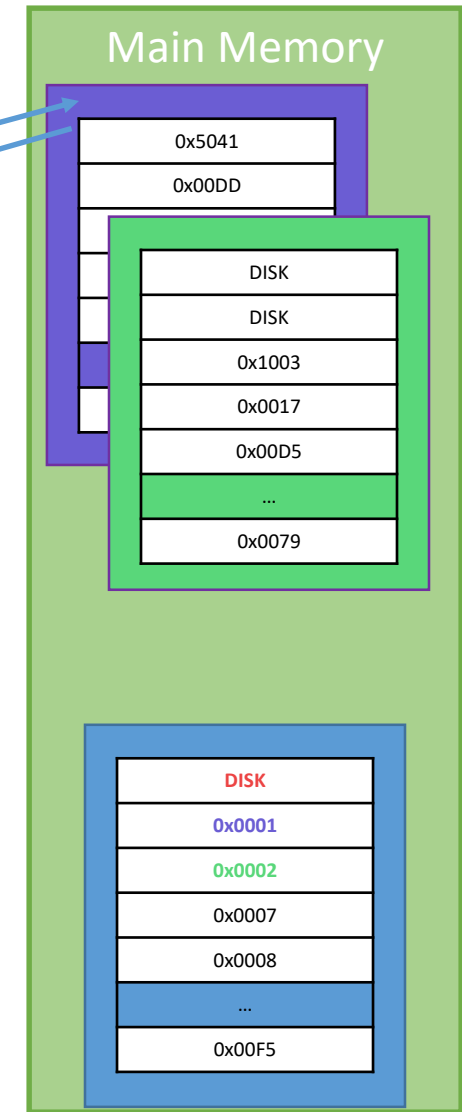
Virtual Address [32 bit]



Physical Address [28 bits]

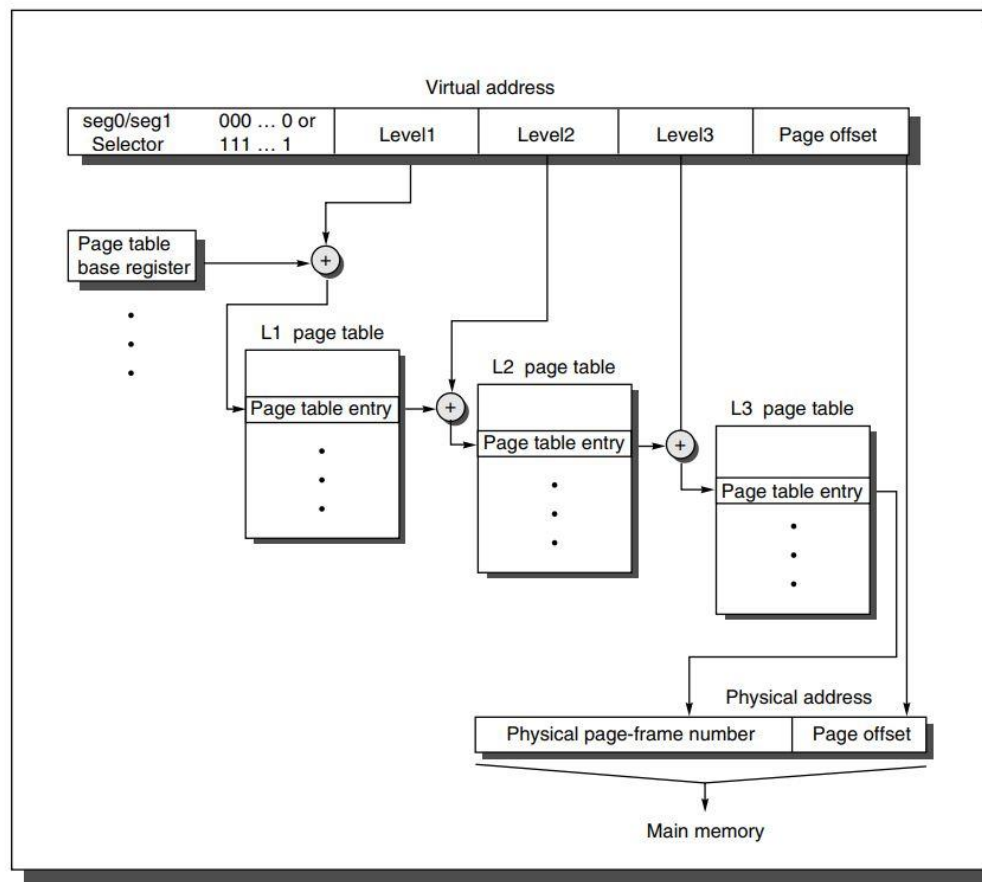


PA: 0x5166 713



Quincy Flint

Illustration from the textbook (advanced)



Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory

Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory
- We need at least one 2nd level Page Table to do translations

Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory
- We need at least one 2nd level Page Table to do translations

- 2-Tier translation

Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory
- We need at least one 2nd level Page Table to do translations
- 2-Tier translation
 - Top-most **10** bits used to index **PT1**

Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory
- We need at least one 2nd level Page Table to do translations
- 2-Tier translation
 - Top-most **10** bits used to index **PT1**
 - Middle-most **10** bits used to index **PT2**

Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory
- We need at least one 2nd level Page Table to do translations
- 2-Tier translation
 - Top-most **10** bits used to index **PT1**
 - Middle-most **10** bits used to index **PT2**
- Advantage:
 - We can now keep Page Tables on disk

Quincy Flint

Multi-Level Page Table: Outro

- We always need to keep the 1st level Page Table in memory
- We need at least one 2nd level Page Table to do translations
- 2-Tier translation
 - Top-most **10** bits used to index **PT1**
 - Middle-most **10** bits used to index **PT2**
- Advantage:
 - We can now keep Page Tables on disk
 - We can still address the same amount of data

Quincy Flint

Quiz: Multi-Level Page Tables

Q: With multilevel page tables, if I am running *100 applications concurrently*, how much memory do I need in RAM?

- I. 4 kB
- II. 8 kB
- III. 8 MB
- IV. 800 kB
- V. 400 MB

Quincy Flint

Quiz: Multi-Level Page Tables

Q: With multilevel page tables, if I am running *100 applications concurrently*, how much memory do I need in RAM?

- I. 4 kB
- II. 8 kB
- III. 8 MB
- IV. 800 kB
- V. 400 MB

A:

- IV. 800 kB

For each program, we must always keep the 1st level page table in RAM (4 kB) and we need at least one 2nd level page table to address data.

Quincy Flint

Quiz: Multi-Level Page Tables

Q: Without multilevel page tables, if I am running *100 applications concurrently*, how much memory do I need in RAM?

- I. 4 kB
- II. 8 kB
- III. 8 MB
- IV. 800 kB
- V. 400 MB

Quincy Flint

Quiz: Multi-Level Page Tables

Q: Without multilevel page tables, if I am running *100 applications concurrently*, how much memory do I need in RAM?

- I. 4 kB
- II. 8 kB
- III. 8 MB
- IV. 800 kB
- V. 400 MB

A:

- IV. 400 MB

For each program, we must keep the entire 4 MB Page Table in RAM at all times.

Quincy Flint

Inverted Page Tables

Quincy Flint

References Quincy Flint

- David Black-Schaffer: Lecture Series on Virtual Memory
- Patterson, Hennessy: Computer Organization and Design: the Hardware/Software Interface
- Intel Hardware Data-Sheets
- **Linux**: Anatomy of a Program in Memory