Review

1. C programming
2. Data rep.
3. Assembly program
4. Cache memories.

Physical Addressing.

CPU

0x100
0x101

Memory bus

Read
word
at addr 4.

Main Memory

0123456

78
The diagram illustrates the process of reading an address from virtual memory to physical memory using a Memory Management Unit (MMU). It shows the flow from the CPU to the MMU, where the virtual address 0x4100 is read and then mapped to a physical address (PA) through the MMU. The RAM is then accessed using this physical address.
Virtual Memory

```
mov 0x40900001, %eax
```

Virtual addr.

```
MMU
```

```
0x408070a2
```

Physical addr
1. Entire program should be in RAM.

2. Segmentation

```
bytes  code  data  stack  heap
 4      4      4      4
```
Paging

Virtual memory (HDD)

Main Memory

Program → page

PPP

Physic Frames

OS + Hardware

MMU

page size = 4KB
Page table - stored in main memory

<table>
<thead>
<tr>
<th>VPN</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Address translation.

VA <-- MMP --> PA
Actual MM size = 4GB + 1GB

<5GB
`malloc`

```
int *pTv = malloc(sizeof(int) * n);
40 bytes for 10 ints.
```
#include <stdlib.h>

void * malloc (size_t size);

void free (void * pty);

free (pty);
pty = NULL;