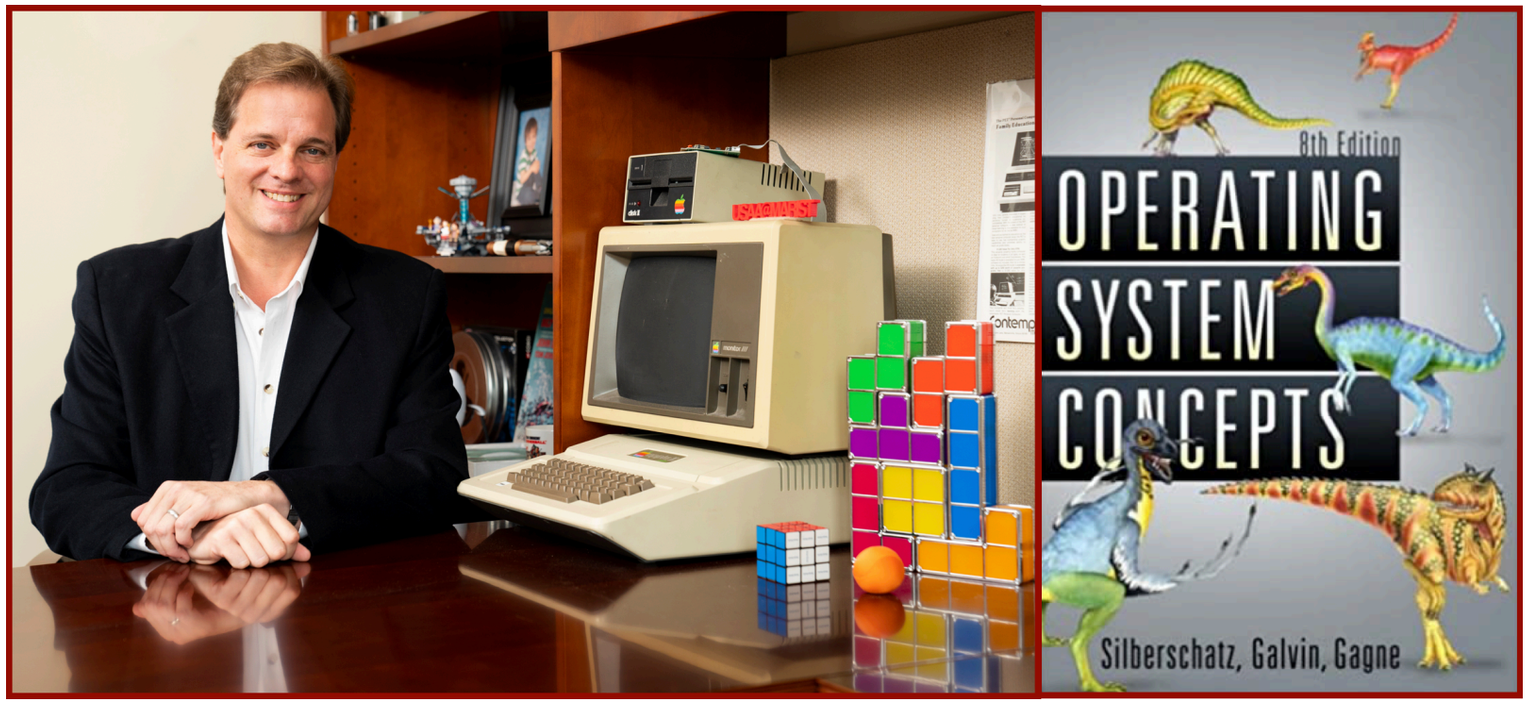
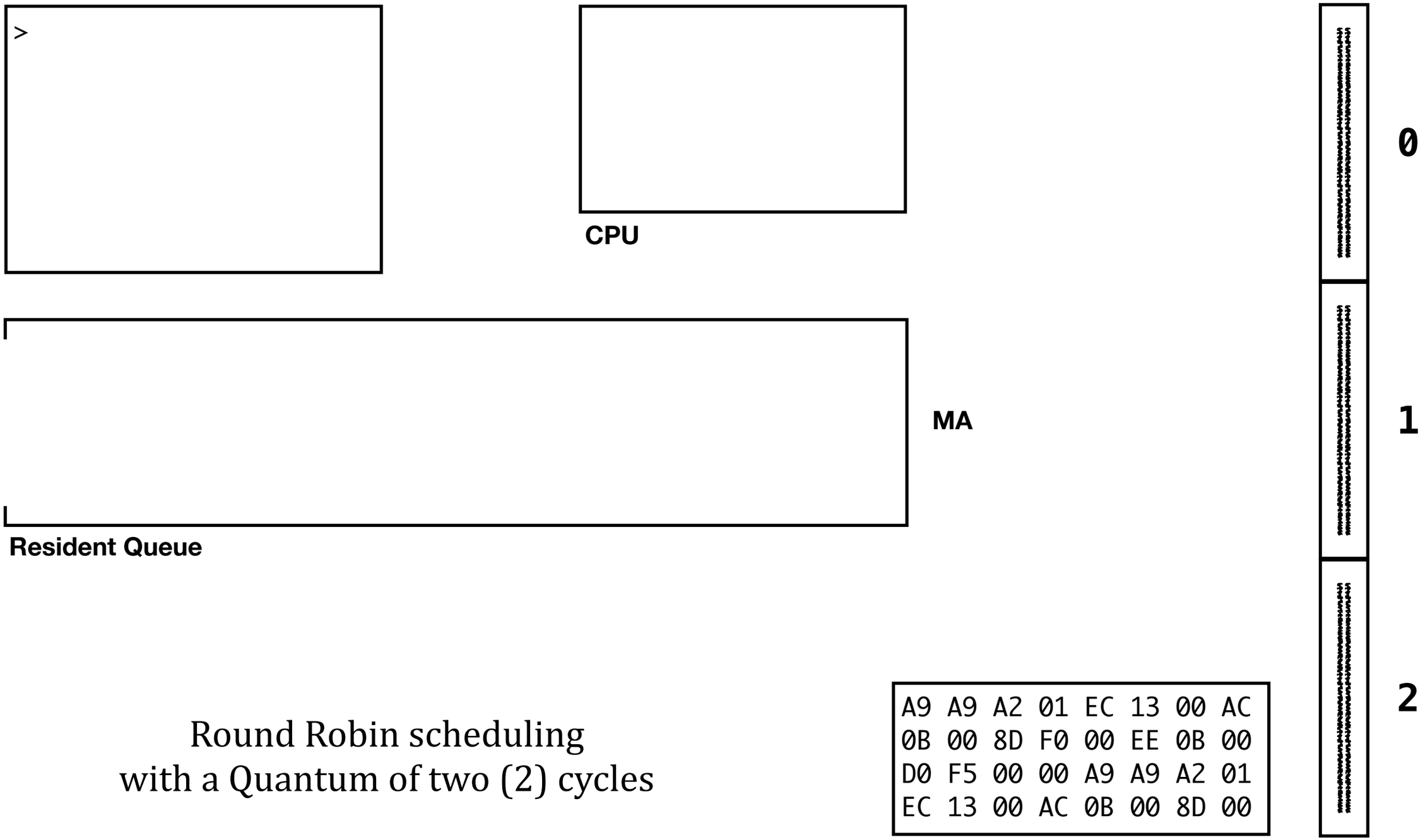
OS *i*Project Three



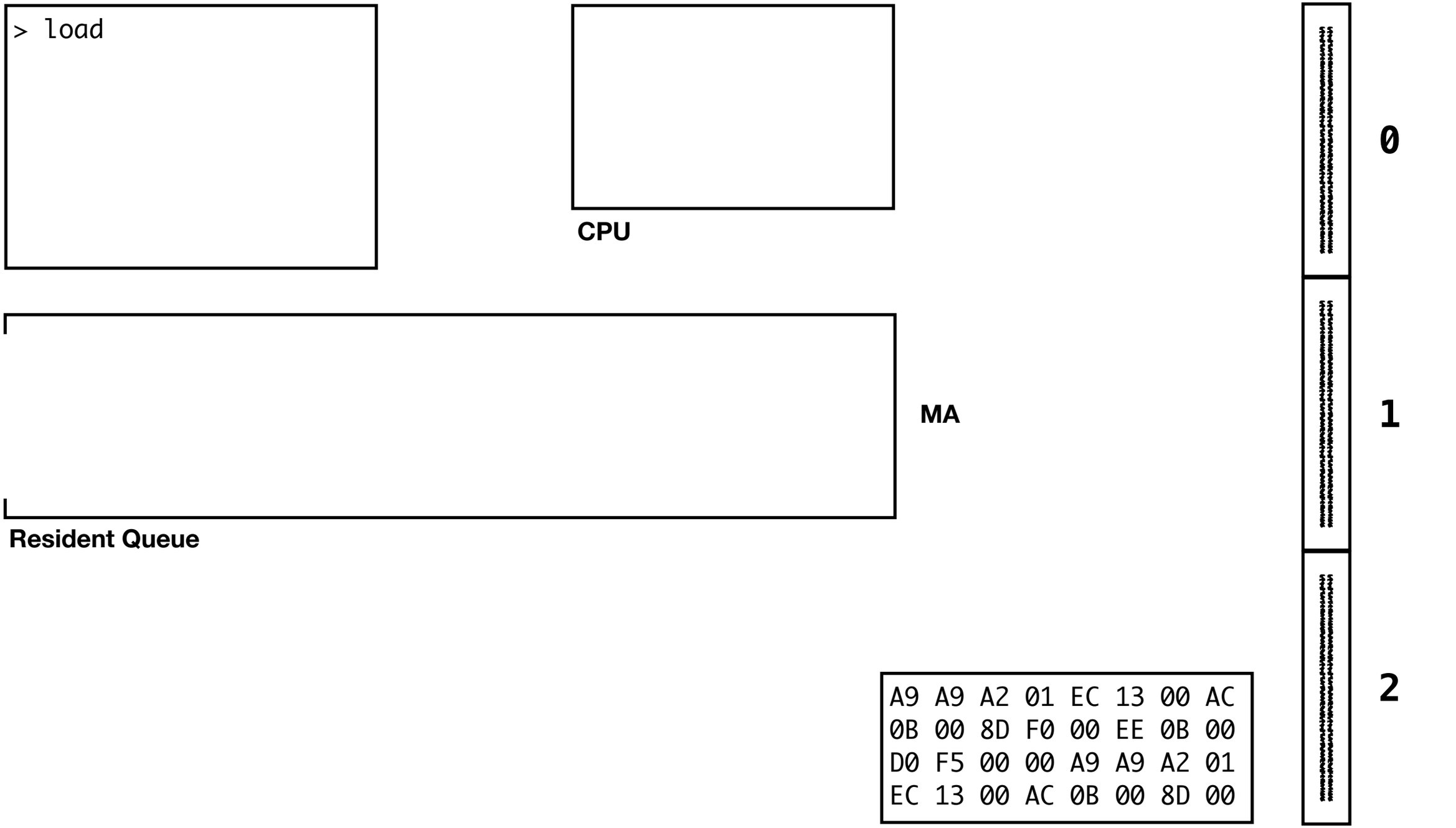
Alan G. Labouseur, Ph.D.
Alan.Labouseur@Marist.edu

*i*Project 3



Round Robin scheduling
with a Quantum of two (2) cycles

*i*Project 3


> load
CPU
MA
Resident Queue
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
0
1
2

*i*Project 3

```
> load
pid 0 loaded in seg 0
>
```

CPU

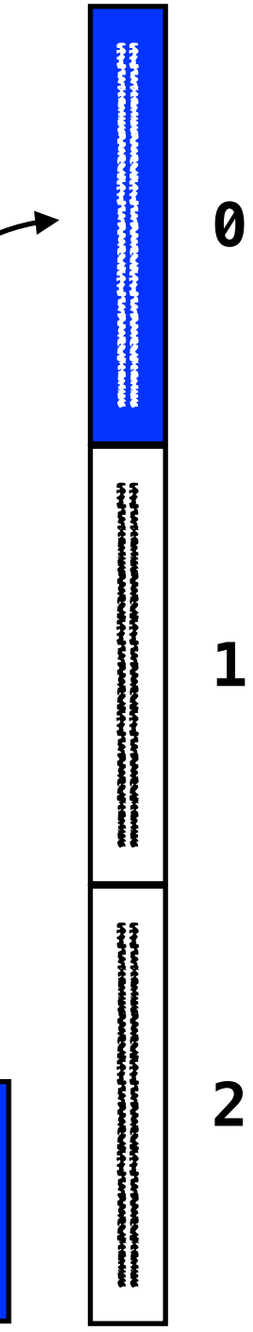
PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

Resident Queue

new PCB()

MMU assigns segment 0, sets base and limit

MA




```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

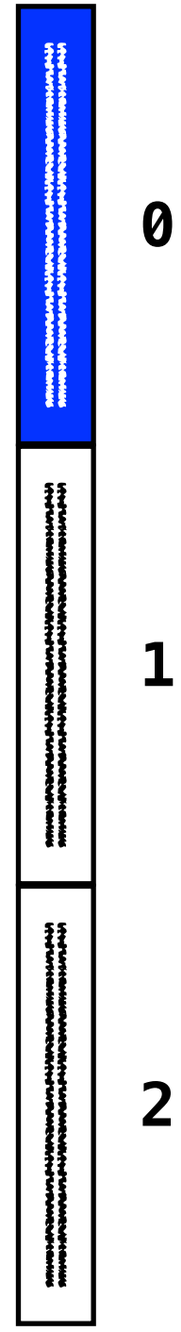
```
> load
pid 0 loaded in seg 0
> load
```

CPU

PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

Resident Queue

MA




```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
>
```

CPU

PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

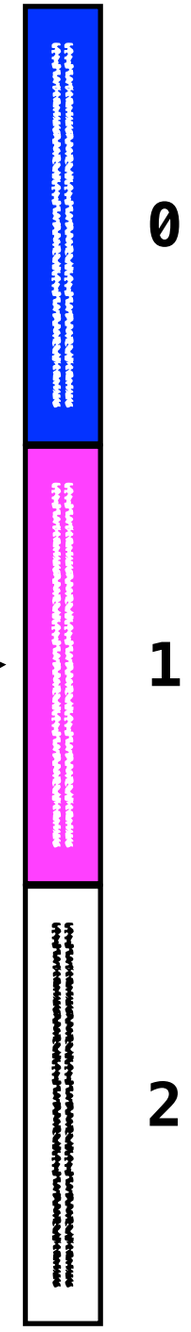
PCB for pid 1
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

Resident Queue

new PCB()

MMU assigns segment 1, sets base and limit

MA




```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
```

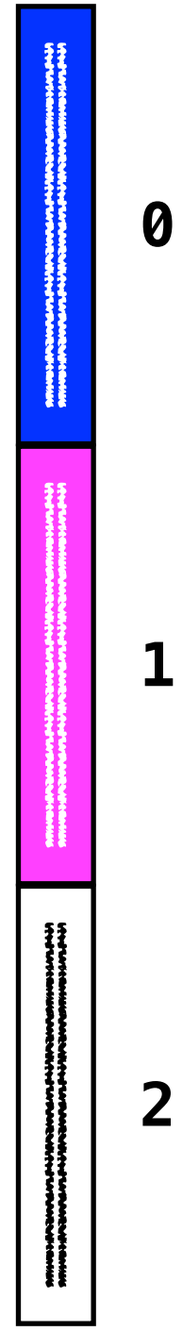
CPU

PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

Resident Queue

MA




```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
>
```

CPU

PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2
Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Resident Queue

MA

0

1

2

A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00

new PCB()

MMU assigns segment 2, sets base and limit

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

CPU

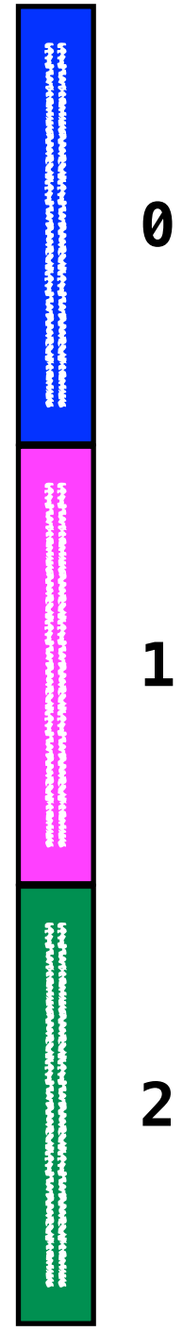
PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2
Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Resident Queue

MA




```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

CPU

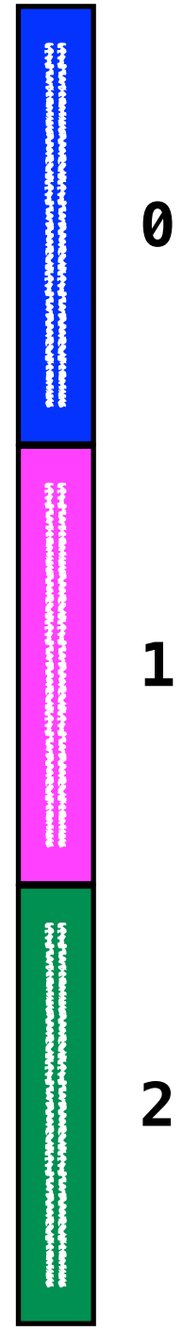
PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2
Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA



Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 0

Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 1

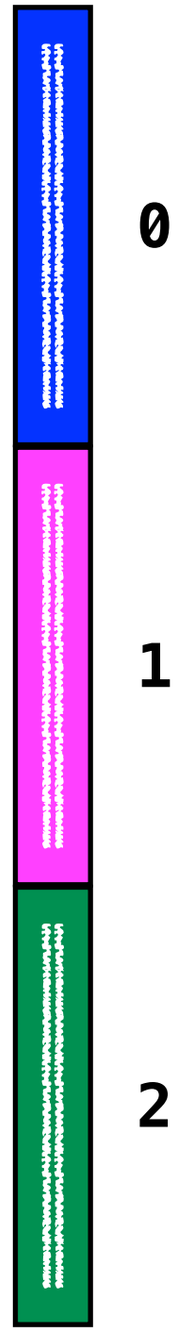
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2

Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA

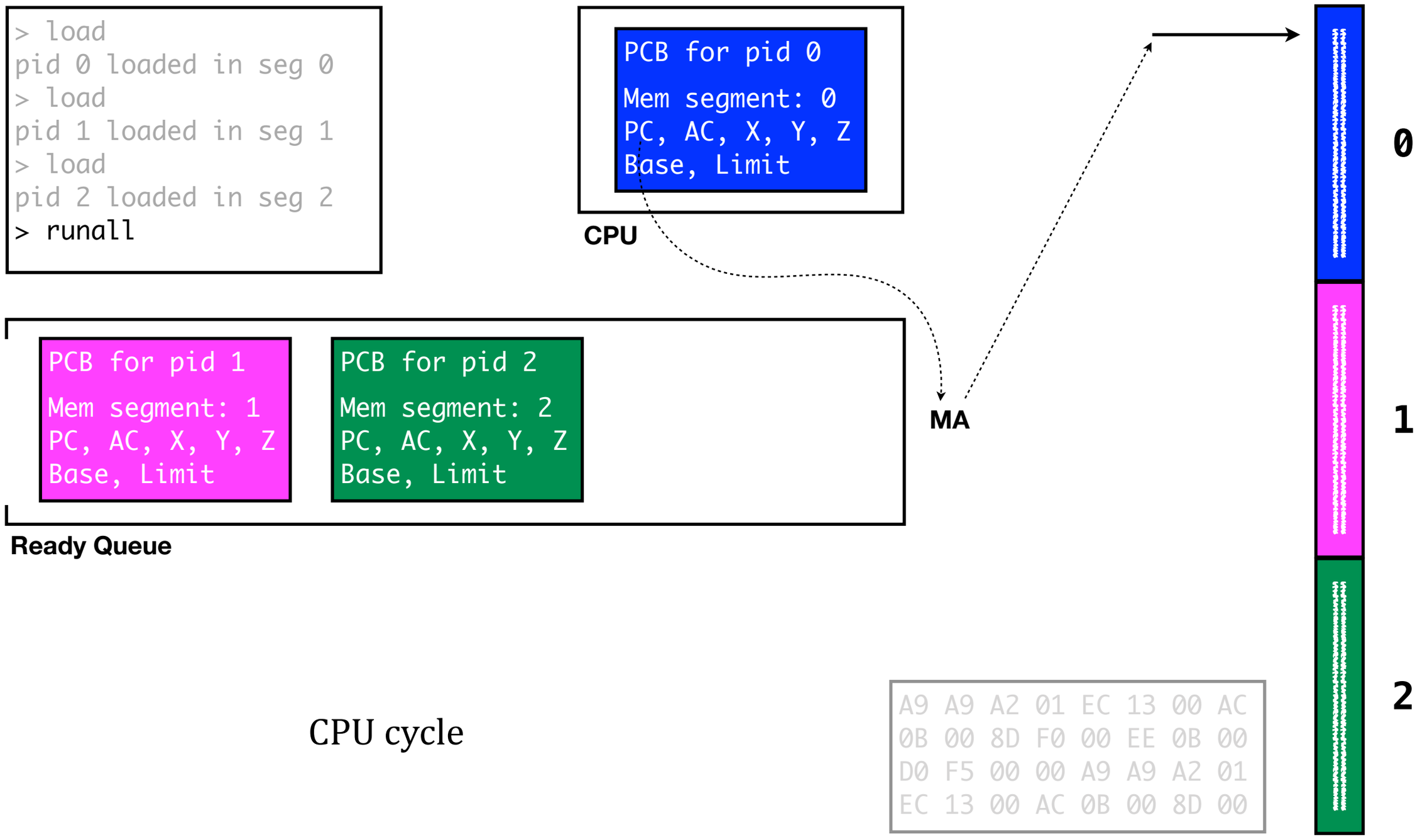


Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

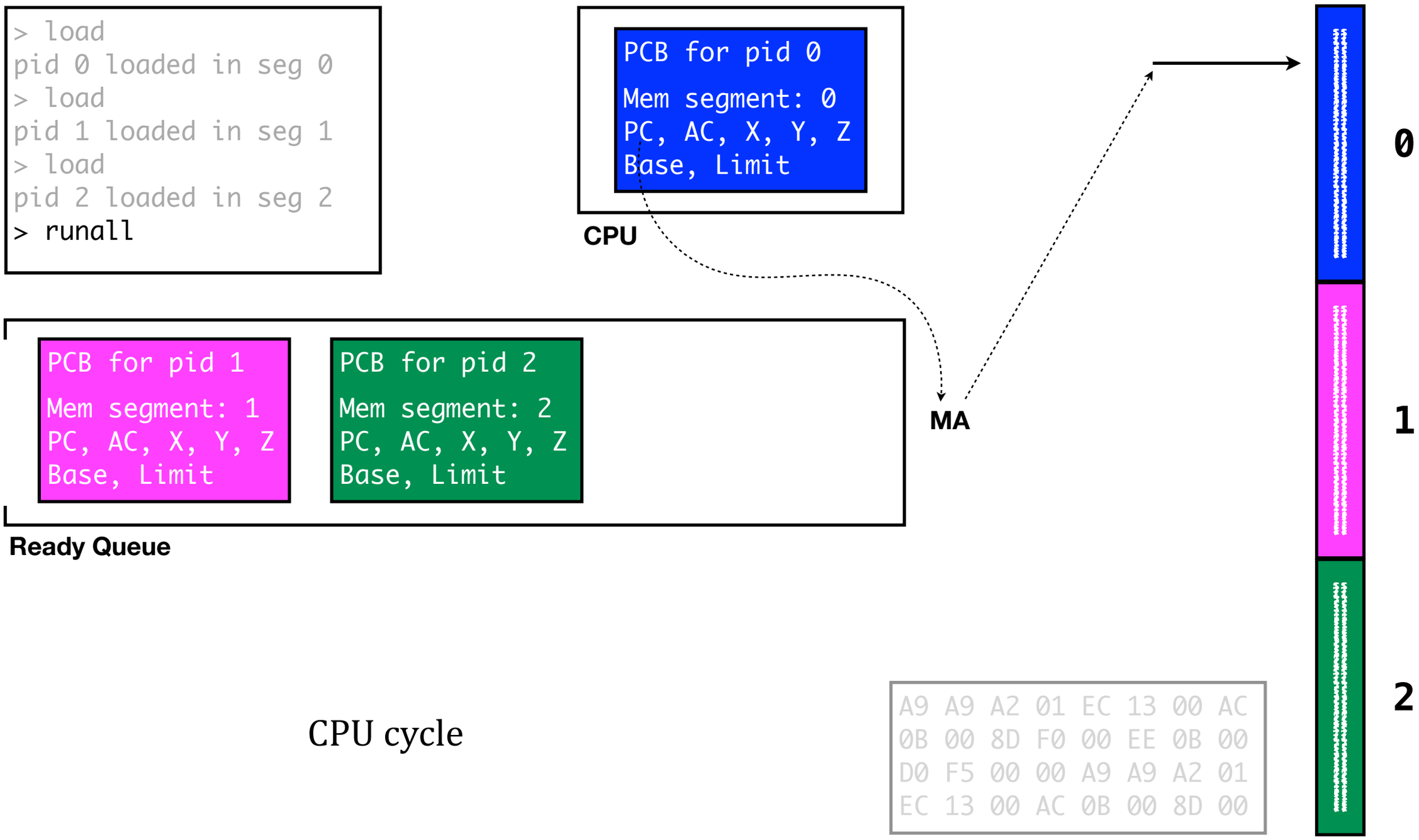


CPU cycle

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```



CPU cycle

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 0

Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 1

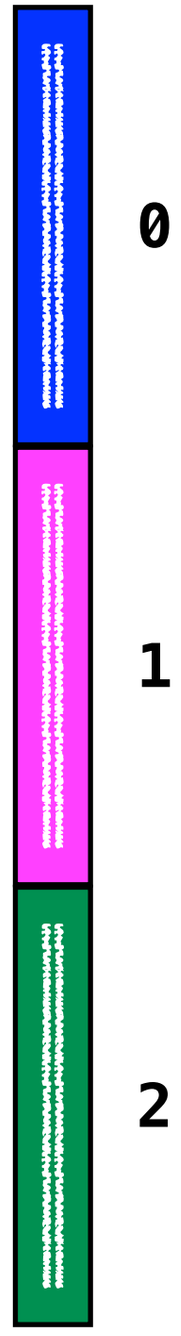
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2

Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA



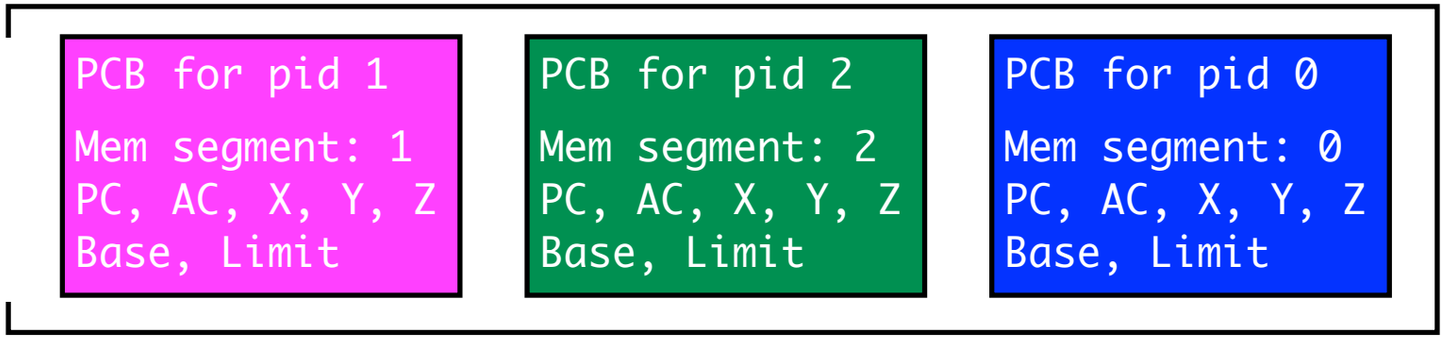
Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

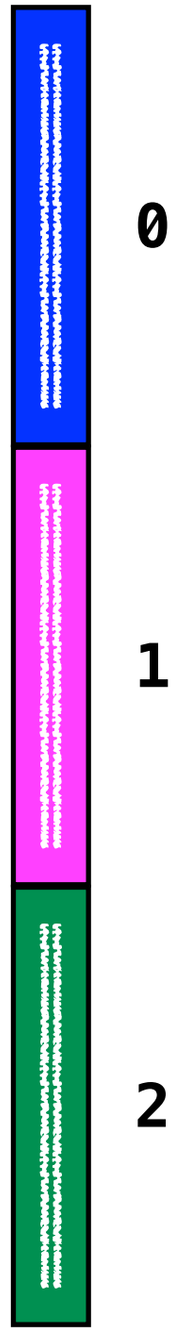
```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

CPU



Ready Queue

MA



Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 1

Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 2

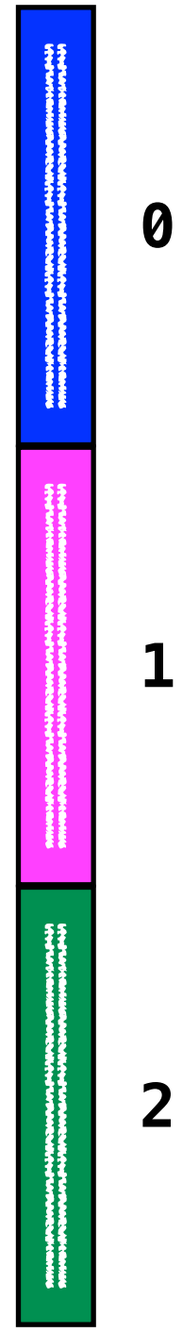
Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

PCB for pid 0

Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA

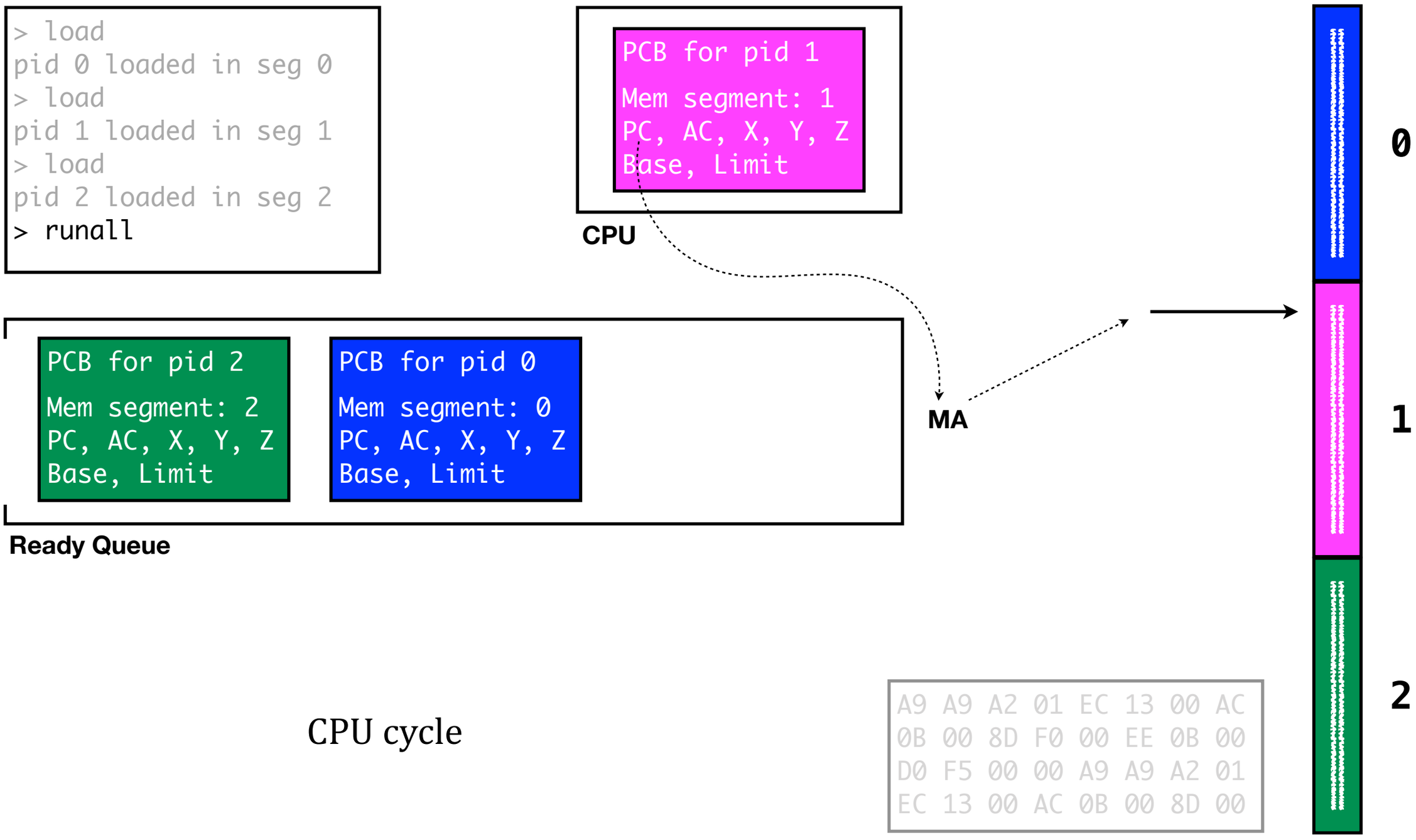


Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

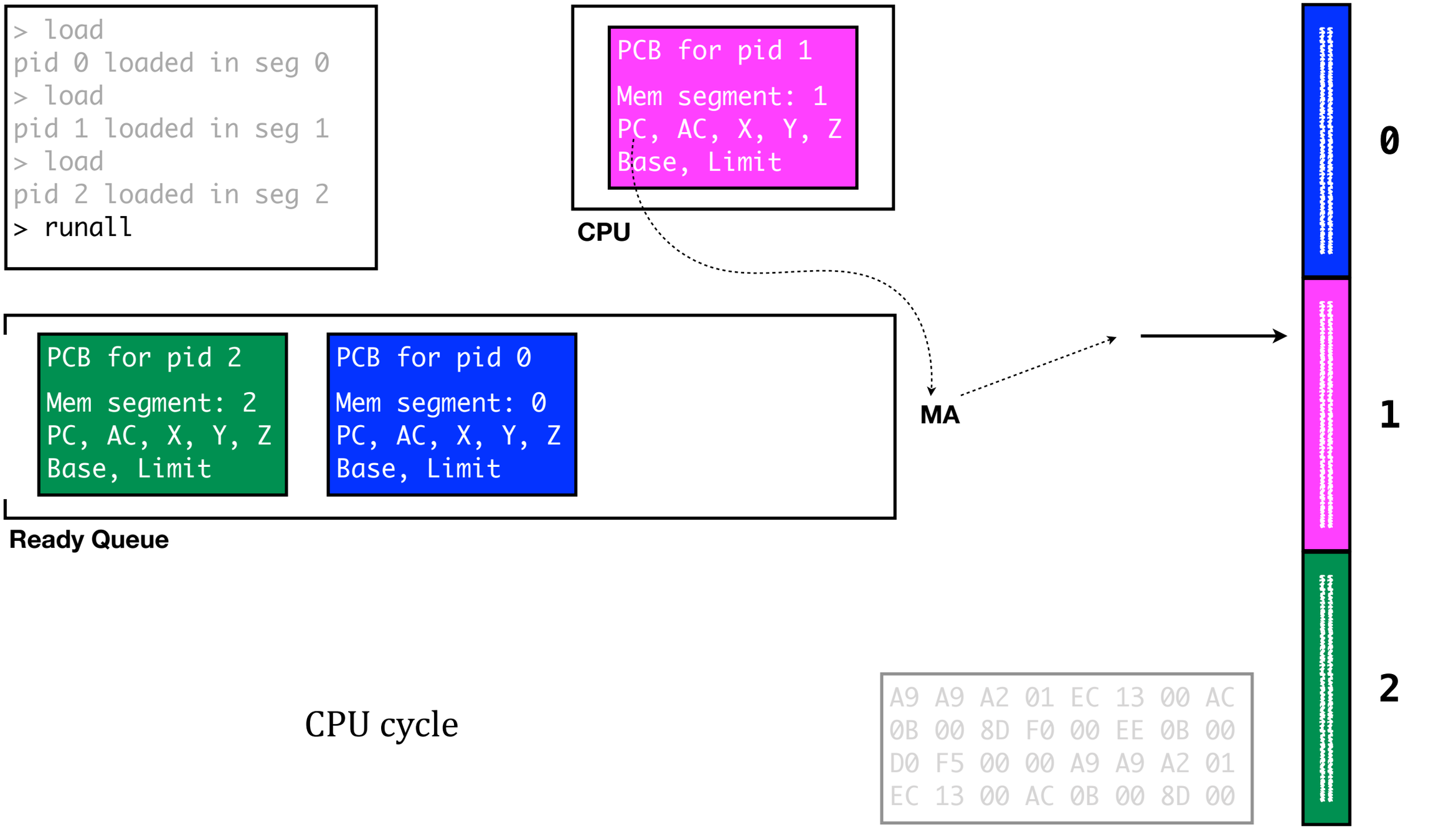


CPU cycle

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```



CPU cycle

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 1

Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 2

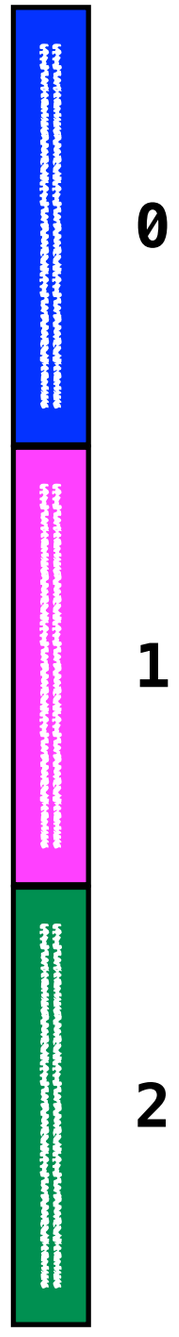
Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

PCB for pid 0

Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA



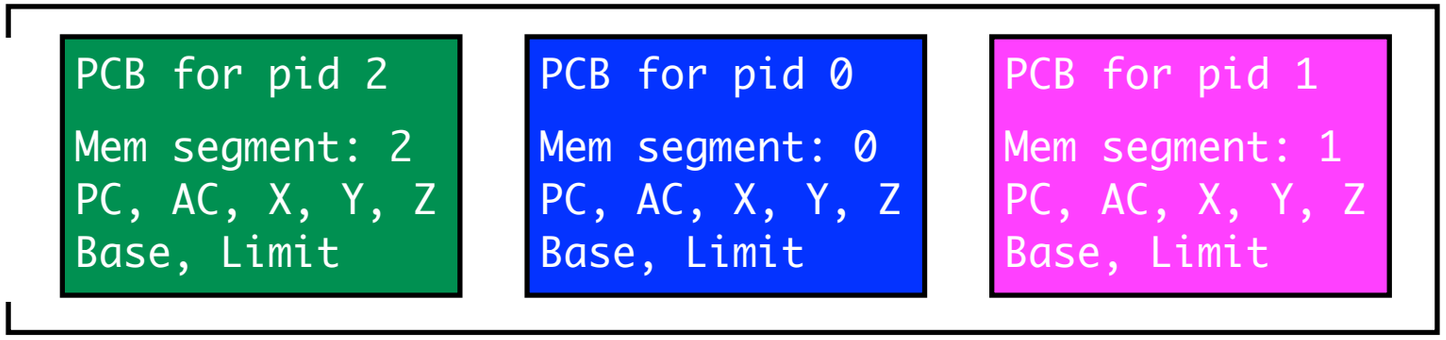
Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

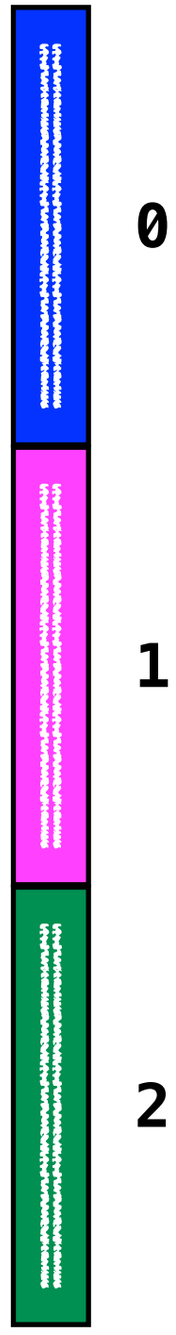
```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

CPU



Ready Queue

MA



Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 2

Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 0

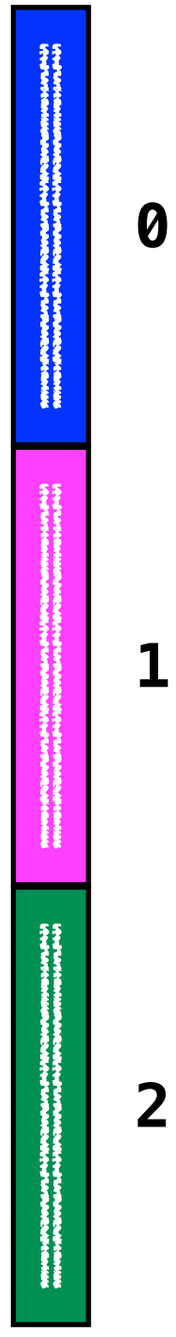
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1

Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA

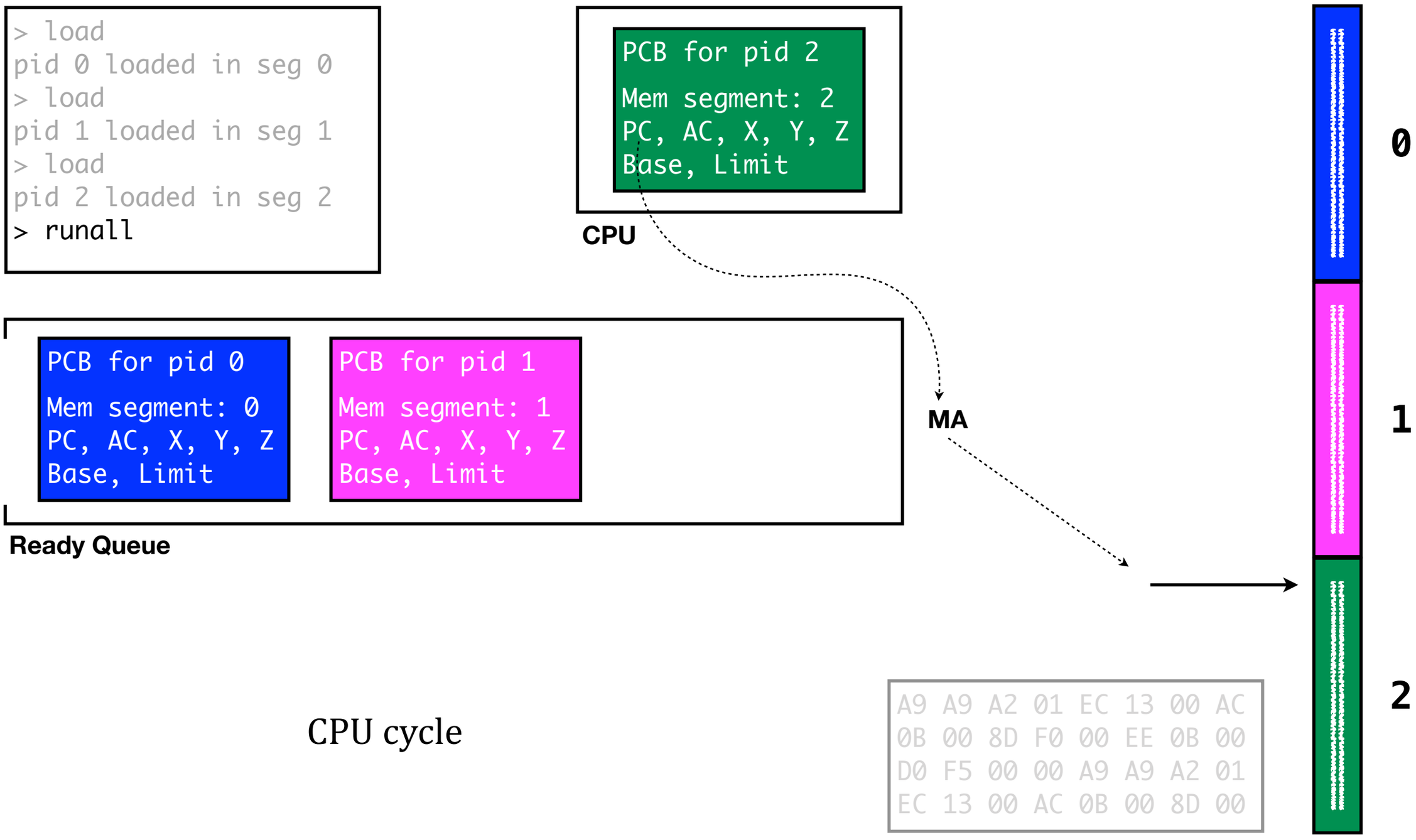


Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

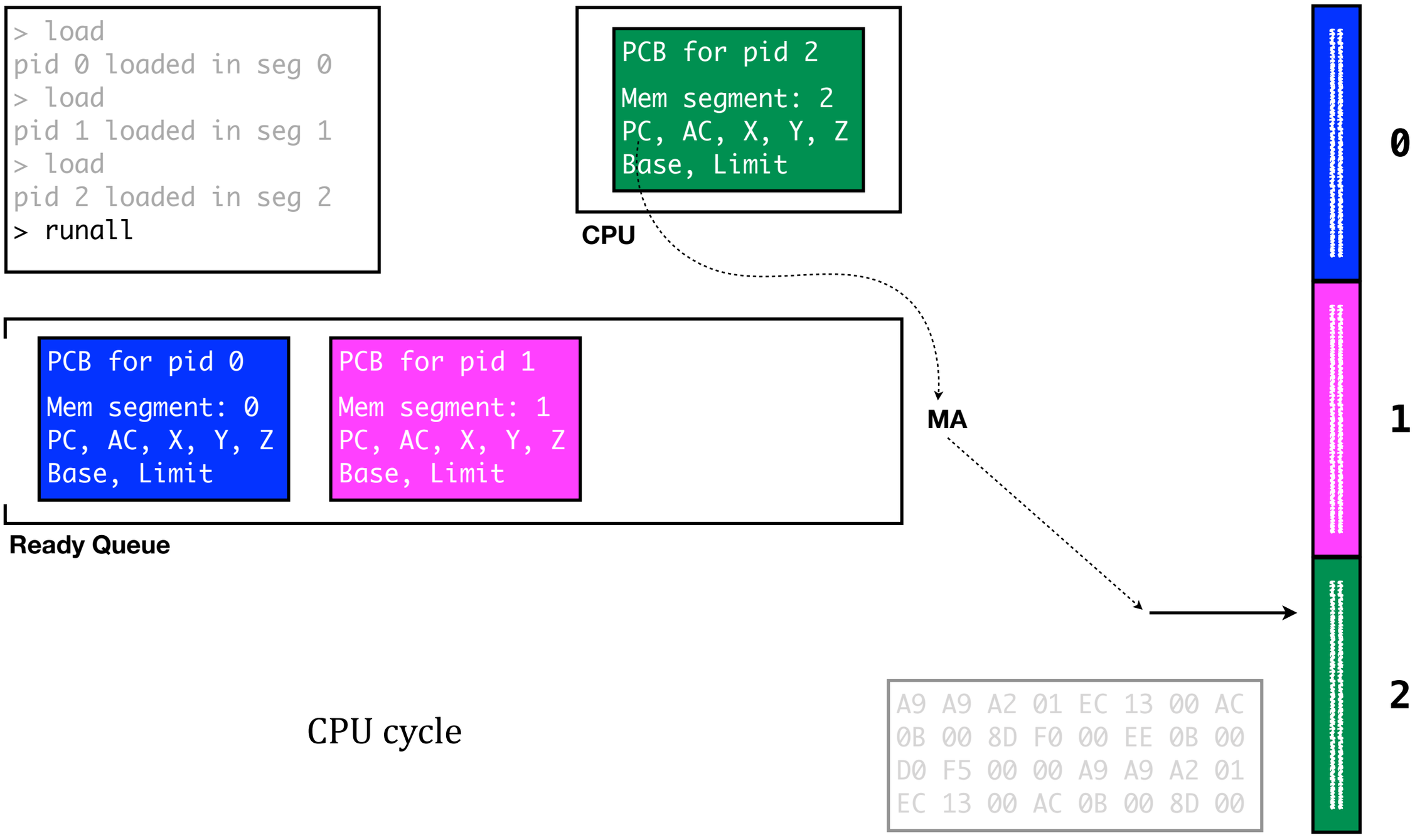


CPU cycle

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```




```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

CPU cycle

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 2

Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 0

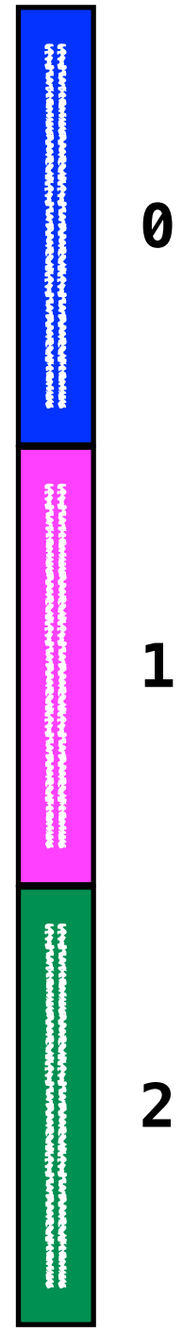
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1

Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA



Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

CPU

PCB for pid 0
Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

PCB for pid 1
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2
Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA

0

1

2

Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

PCB for pid 0

Mem segment: 0
PC, AC, X, Y, Z
Base, Limit

CPU

PCB for pid 1

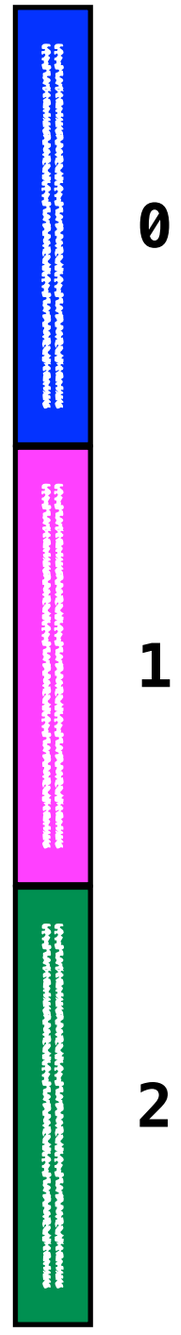
Mem segment: 1
PC, AC, X, Y, Z
Base, Limit

PCB for pid 2

Mem segment: 2
PC, AC, X, Y, Z
Base, Limit

Ready Queue

MA

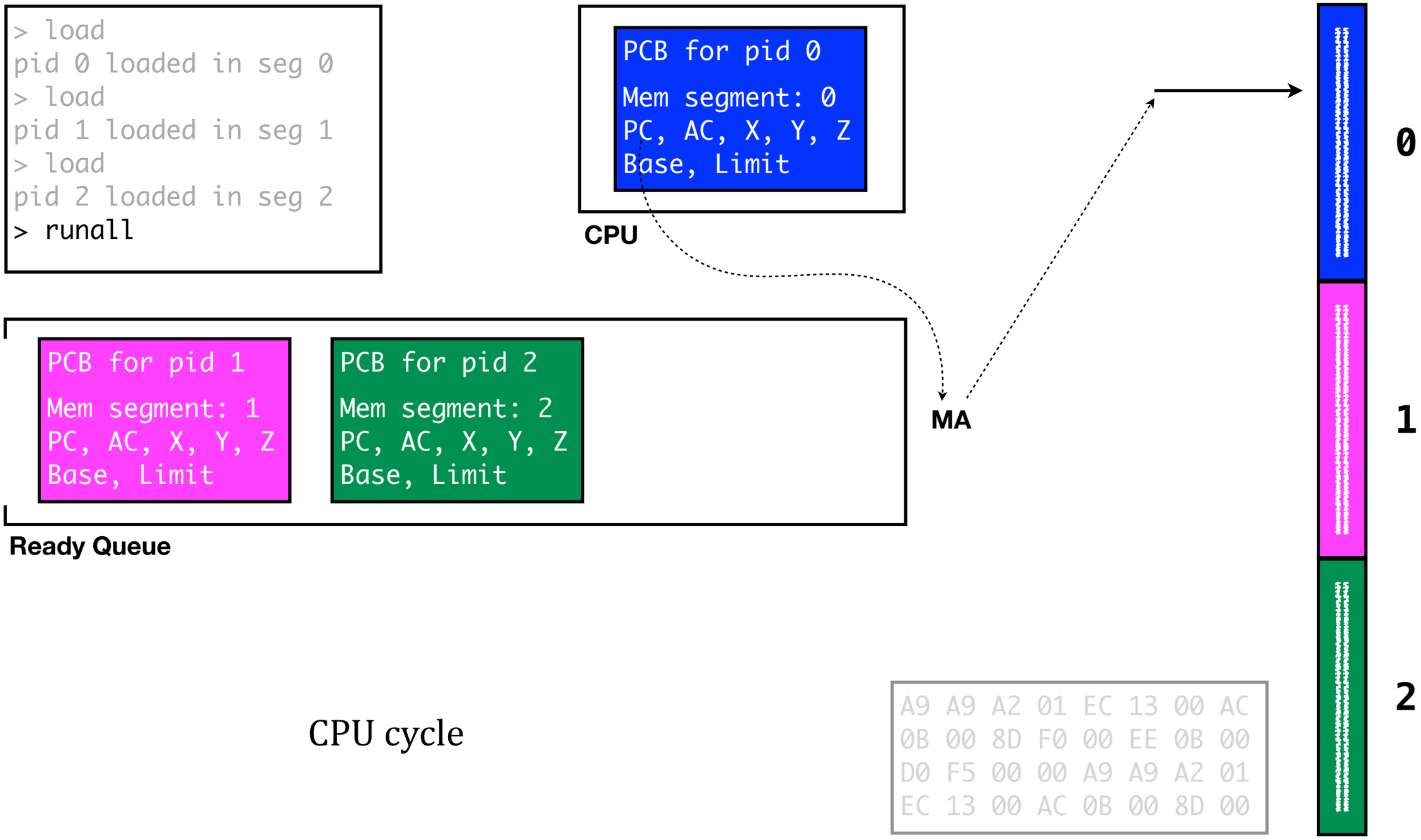


Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

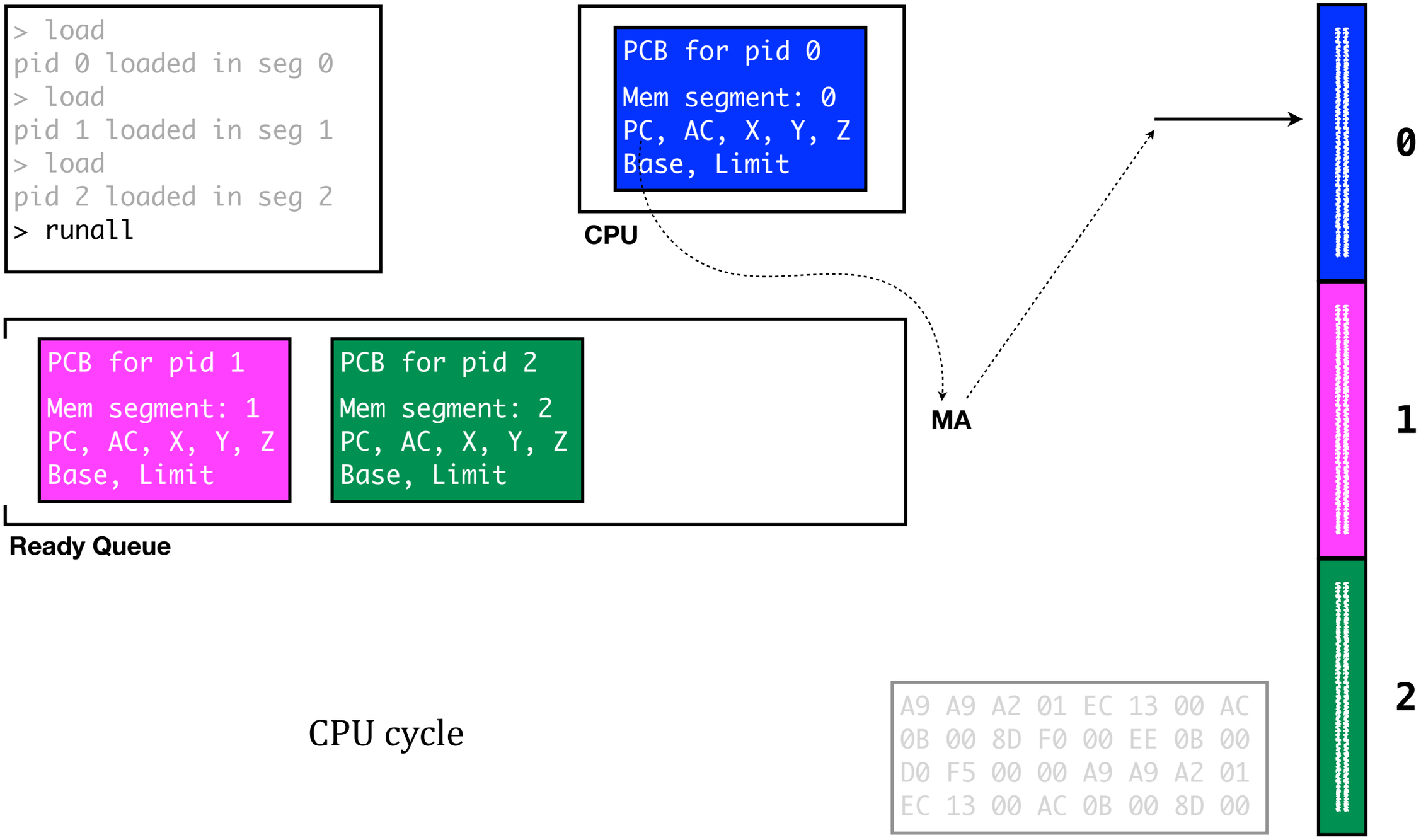


CPU cycle

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

iProject 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```

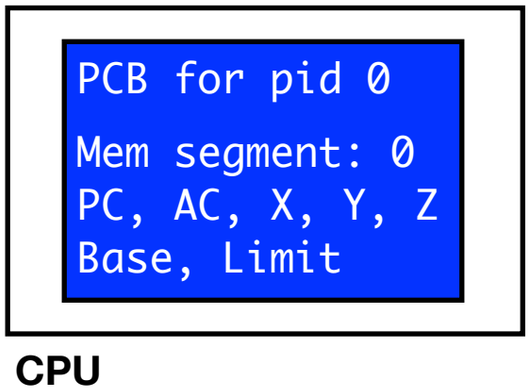


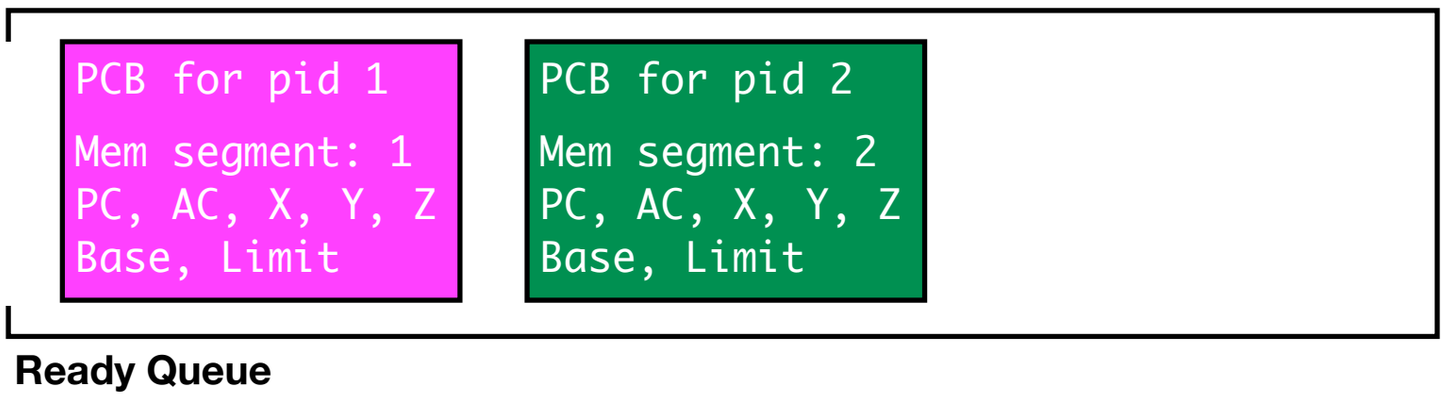
CPU cycle

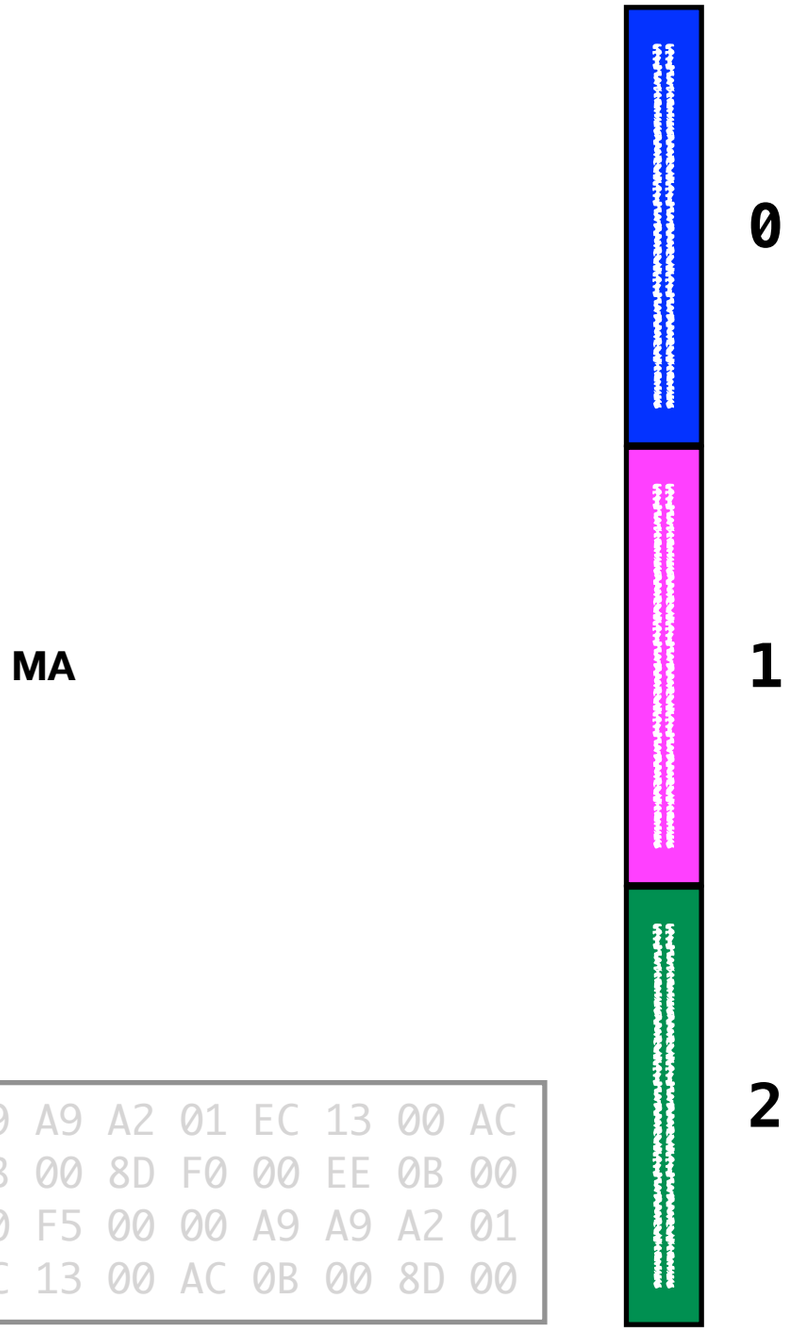
```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

*i*Project 3

```
> load
pid 0 loaded in seg 0
> load
pid 1 loaded in seg 1
> load
pid 2 loaded in seg 2
> runall
```







Context Switch

```
A9 A9 A2 01 EC 13 00 AC
0B 00 8D F0 00 EE 0B 00
D0 F5 00 00 A9 A9 A2 01
EC 13 00 AC 0B 00 8D 00
```

. . . and so on, until the processes are complete.