

SCS1312 Operating System Concepts

Dr. Chamath Keppitiyagama

University of Colombo School of Computing

fork()

fork()

```
#include <unistd.h>
#include <stdio.h>

int main()
{
    int x=1;

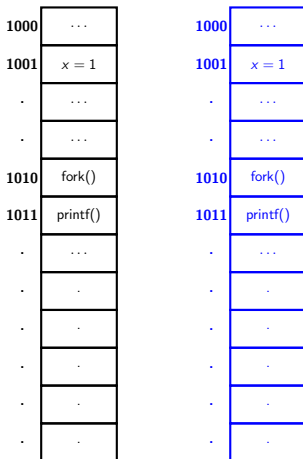
    fork();

    printf("%d\n", x);
}
```

fork()

1000	...
1001	$x = 1$
2	...
.	...
1010	fork()
1011	printf()
.	...
.	.
.	.
.	.
.	.
.	.
.	.

fork()



fork()

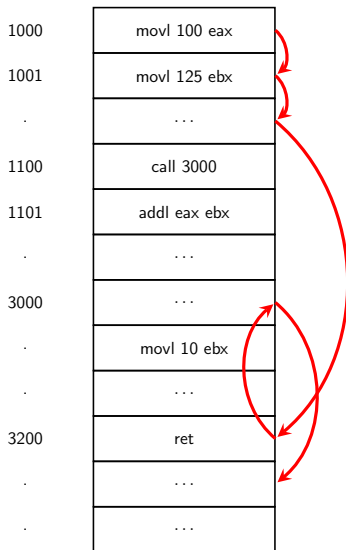
```
#include <unistd.h>
#include <stdio.h>

int main()
{
    int x=100;

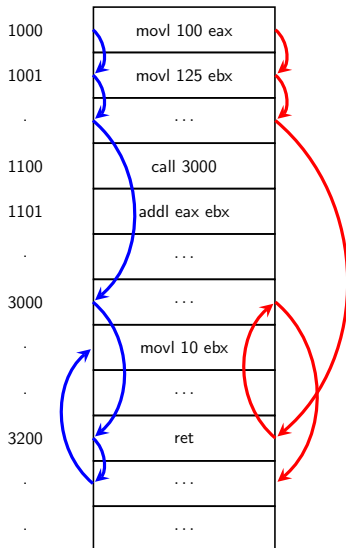
    if(fork())
        printf("%d\n", x);
    else
        printf("Child\n");
}
```

1000	movl 100 eax
1001	movl 125 ebx
.	...
1100	call 3000
1101	addl eax ebx
.	...
3000	...
.	movl 10 ebx
.	...
3200	ret
.	...
.	...





Thread



Payroll

$x = ACC$
$x = x + 1000$
$ACC = x$

--

x

ATM

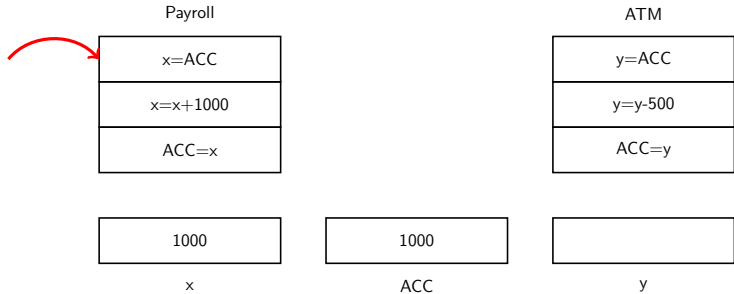
$y = ACC$
$y = y - 500$
$ACC = y$

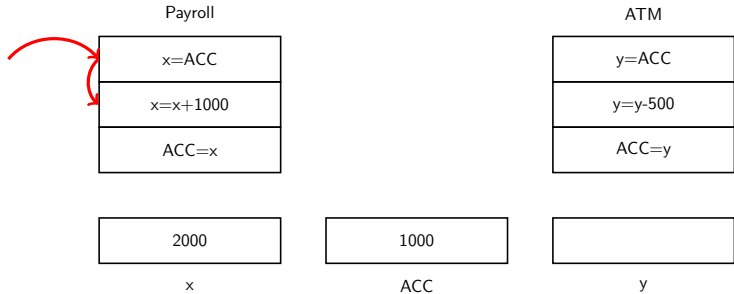
--

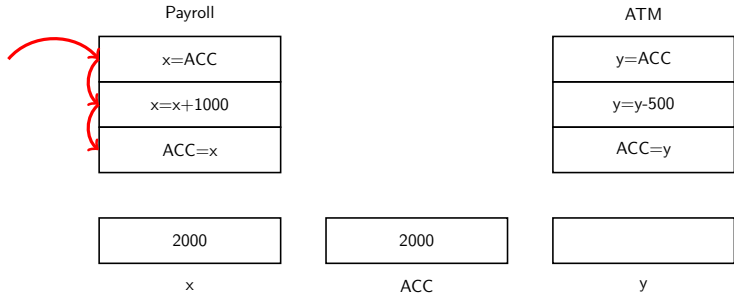
y

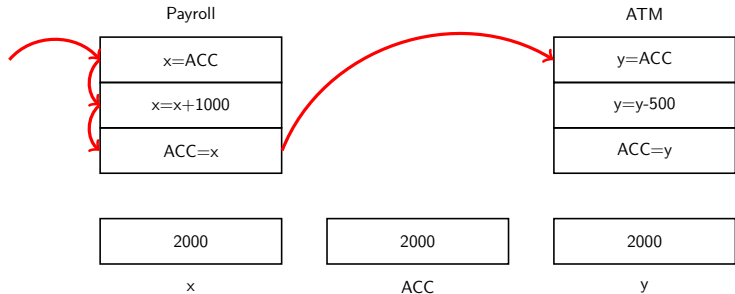
1000

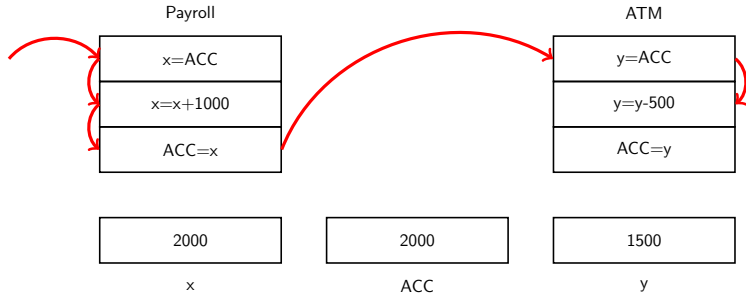
ACC

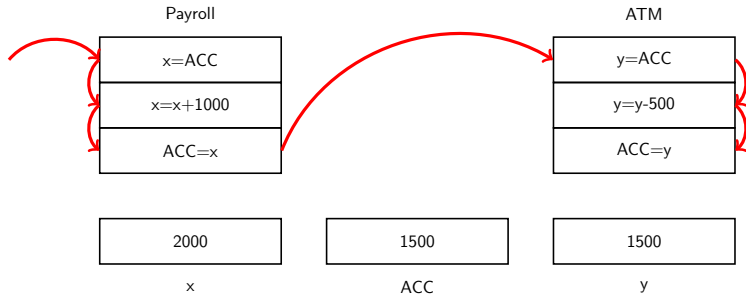




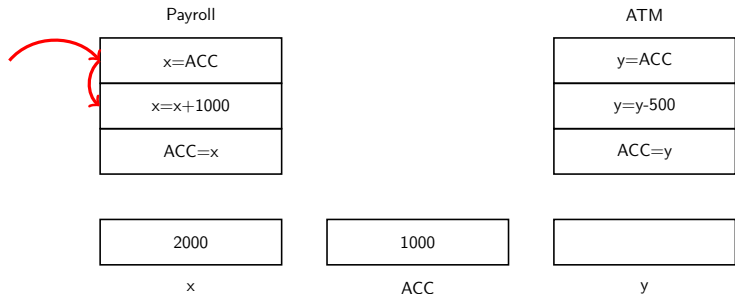




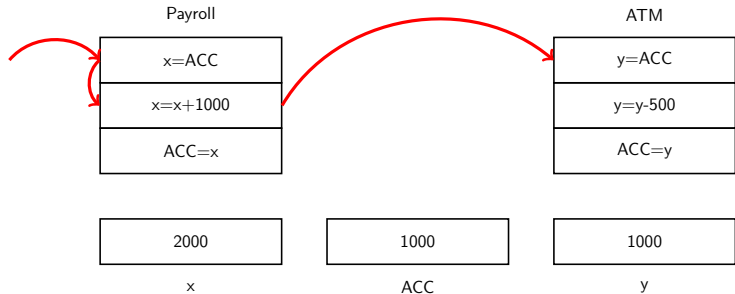




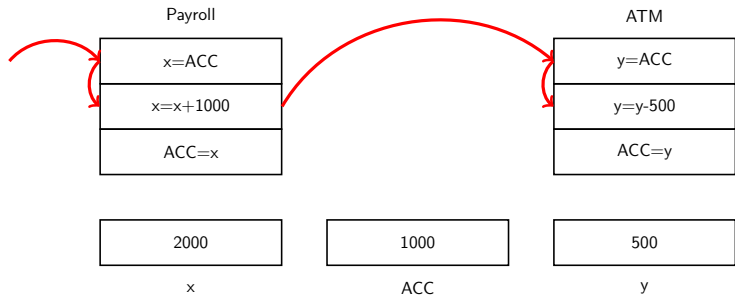
Another Path



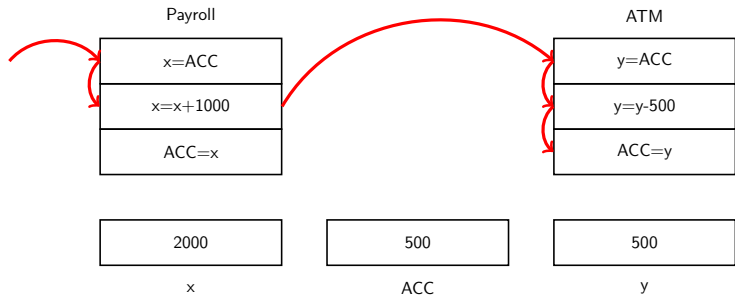
Another Path



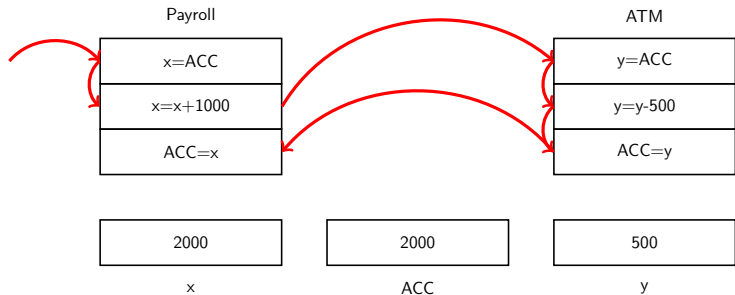
Another Path



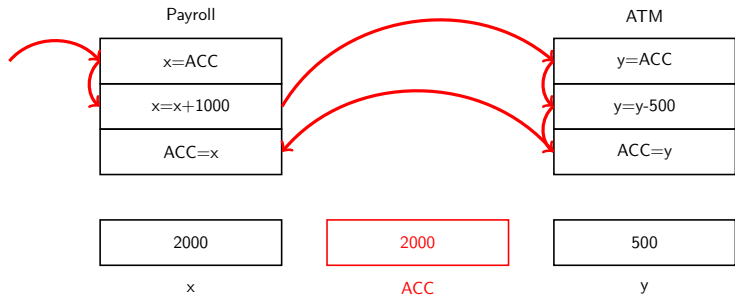
Another Path



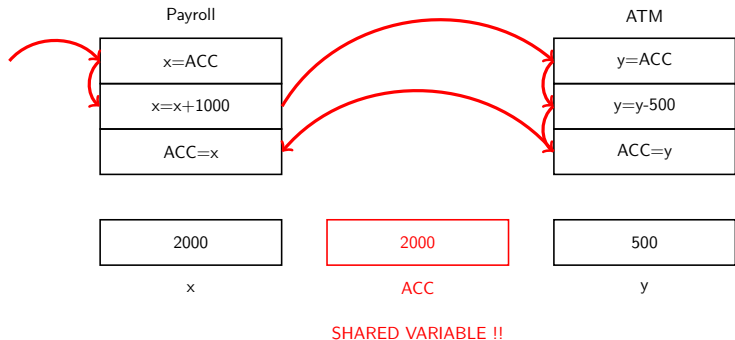
Another Path



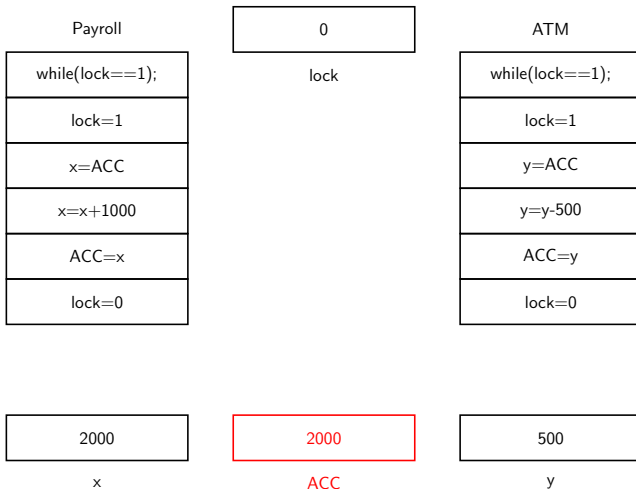
Another Path



Another Path

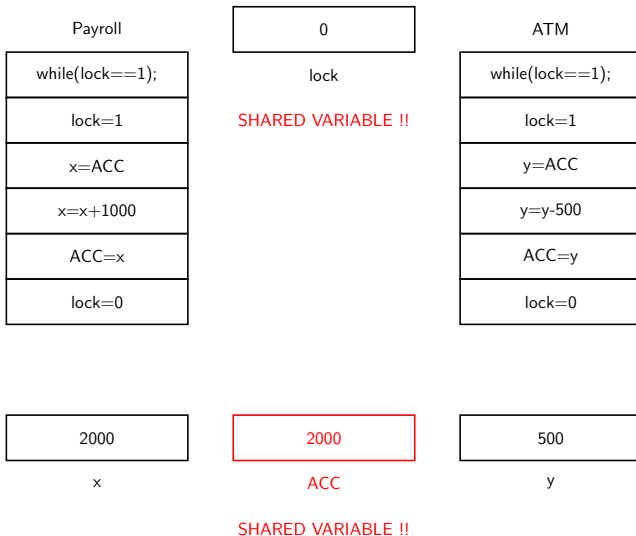


Using a Lock



SHARED VARIABLE !!

Using a Lock



- Race Condition
- Critical Section Problem

- Atomicity

General Structure

```
noncritical_section();
```

```
critical_section();
```

```
noncritical_section();
```

Test_And_Set(x)

Returns the current value of x and the sets the value of x to *True*.