

CMSC 313
Introduction to Computer Systems
Lecture 15
Assembly Language, cont.
and Process Control

Alan Sussman
als@cs.umd.edu

Administrivia

- Project 4 spec and public tests posted
- Finish reading Bryant and O'Hallaron Section 4.1 (Y86 subset) and Chapter 3, for more info on IA-32 instruction set architecture
- Start reading Sections 8.2-8.5, on process control

Chapters 3 and 4.1, Bryant and O'Hallaron

ASSEMBLY LANGUAGE

Final Factorial Program

```
0x000: |          rdint  %ecx
0x002: |          irmovl $1,%ebx
0x008: |          irmovl $1,%eax
0x00e: |          rrmovl %eax,%edi
0x010: |          subl  %ecx,%edi
0x012: |          jg     0x2a

0x017: | Loop:      multl  %eax,%ebx
0x019: |          irmovl $1,%edi
0x01f: |          addl  %edi,%eax
0x021: |          rrmovl %eax,%edi
0x023: |          subl  %ecx,%edi
0x025: |          jle   0x17

0x02a: | EndWhile:  wrint  %ebx
0x02c: |          irmovl $10,%edi
0x032: |          wrch  %edi
0x034: |          halt
```

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	0
%ecx	0
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x000
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	0
%ecx	0
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x002
CC: ZF	0
CC: SF	0
CC: OF	0

Input is: 4

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x002
CC: ZF	0
CC: SF	0
CC: OF	0

Input is: 4

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x008
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x008
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x00e
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x00e
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x010
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x010
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x012
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x012
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

jg:
 $\sim(SF \wedge OF) \ \& \ \sim ZF = 0$

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	2

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	2

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x025
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x02a
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

jle:
(SF ^ OF) | ZF = 1

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt

```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	3

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	3

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x025
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x02a
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

jle:
(SF ^ OF) | ZF = 1

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	4

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	4

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x025
CC: ZF	1
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x02a
CC: ZF	1
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

jle:
(SF ^ OF) | ZF = 1

PC	0x017
CC: ZF	1
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl  %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl  %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl   %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl   %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	5

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl   %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl   %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	5

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl   %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl   %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint  %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl   %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl   %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x02a
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

jle:
(SF ^ OF) | ZF = 0

PC	0x02a
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x02c
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

Output is: 24

PC	0x02c
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

Output is: 24

PC	0x032
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

0x000:	rdint %ecx		
0x002:	irmovl \$1,%ebx		
0x008:	irmovl \$1,%eax		
0x00e:	rrmovl %eax,%edi		
0x010:	subl %ecx,%edi		
0x012:	jg 0x2a		
0x017:	multl %eax,%ebx		
0x019:	irmovl \$1,%edi		
0x01f:	addl %edi,%eax		
0x021:	rrmovl %eax,%edi		
0x023:	subl %ecx,%edi		
0x025:	jle 0x17		
0x02a:	wrint %ebx		
0x02c:	irmovl \$10,%edi		
0x032:	wrch %edi		
0x034:	halt		

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x032
CC: ZF	0
CC: SF	0
CC: OF	0

Output is: 24

Execution

0x000:	rdint %ecx		
0x002:	irmovl \$1,%ebx		
0x008:	irmovl \$1,%eax		
0x00e:	rrmovl %eax,%edi		
0x010:	subl %ecx,%edi		
0x012:	jg 0x2a		
0x017:	multl %eax,%ebx		
0x019:	irmovl \$1,%edi		
0x01f:	addl %edi,%eax		
0x021:	rrmovl %eax,%edi		
0x023:	subl %ecx,%edi		
0x025:	jle 0x17		
0x02a:	wrint %ebx		
0x02c:	irmovl \$10,%edi		
0x032:	wrch %edi		
0x034:	halt		

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x034
CC: ZF	0
CC: SF	0
CC: OF	0

Output is: 24

Execution

0x000:	rdint %ecx		
0x002:	irmovl \$1,%ebx		
0x008:	irmovl \$1,%eax		
0x00e:	rrmovl %eax,%edi		
0x010:	subl %ecx,%edi		
0x012:	jg 0x2a		
0x017:	multl %eax,%ebx		
0x019:	irmovl \$1,%edi		
0x01f:	addl %edi,%eax		
0x021:	rrmovl %eax,%edi		
0x023:	subl %ecx,%edi		
0x025:	jle 0x17		
0x02a:	wrint %ebx		
0x02c:	irmovl \$10,%edi		
0x032:	wrch %edi		
0x034:	halt		

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x034
CC: ZF	0
CC: SF	0
CC: OF	0

Output is: 24

Execution

0x000:	rdint %ecx		
0x002:	irmovl \$1,%ebx		
0x008:	irmovl \$1,%eax		
0x00e:	rrmovl %eax,%edi		
0x010:	subl %ecx,%edi		
0x012:	jg 0x2a		
0x017:	multl %eax,%ebx		
0x019:	irmovl \$1,%edi		
0x01f:	addl %edi,%eax		
0x021:	rrmovl %eax,%edi		
0x023:	subl %ecx,%edi		
0x025:	jle 0x17		
0x02a:	wrint %ebx		
0x02c:	irmovl \$10,%edi		
0x032:	wrch %edi		
0x034:	halt		

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x035
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```

0x000: | rdint %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl %ecx,%edi
0x012: | jg 0x2a
0x017: | multl %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl %ecx,%edi
0x025: | jle 0x17
0x02a: | wrint %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch %edi
0x034: | halt
    
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x035
CC: ZF	0
CC: SF	0
CC: OF	0

Done!

Execution

- These are the final states of the PC, CC, and all the registers.
- Does this agree with what **yis** tells us?
- Remember, numbers are printed in hex, so $10_{10} = a_{16}$, $24_{10} = 18_{16}$

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x035
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

- Yes:

```

$ echo 4 | yis factorial.yo
24
    
```

Stopped in 34 steps at PC = 0x35. Exception 'HLT', CC Z=0 S=0 O=0
Changes to registers:

```

%eax: 0x00000000    0x00000005
%ecx: 0x00000000    0x00000004
%ebx: 0x00000000    0x00000018
%edi: 0x00000000    0x0000000a
    
```

Changes to memory:

Some other notes

- Exceptions
 - If your code exits due to a **halt** instruction, the exception will be HLT (which is fine)
 - You can also stop your code prematurely by passing a second parameter (max steps) to **yis**; this will usually result in the exception AOK
 - But if you manage to move to a bad address (forget to halt?), divide by zero, or just make the simulator otherwise unhappy, you'll get a different exception (e.g., ADR or INS)

Program efficiency

- Our assembly program was not as efficient as it could be
 - we could have optimized away the need for the variable **i** by simply decrementing **n** until **n** was 0
 - we also could have been smarter with our registers, and not overwritten the constant value we used repeatedly (1)
- These things need to be taken into consideration by compilers, and by you in the assembly code you're writing for the project...

Sections 8.2-8.5, Bryant and O'Hallaron

PROCESS CONTROL

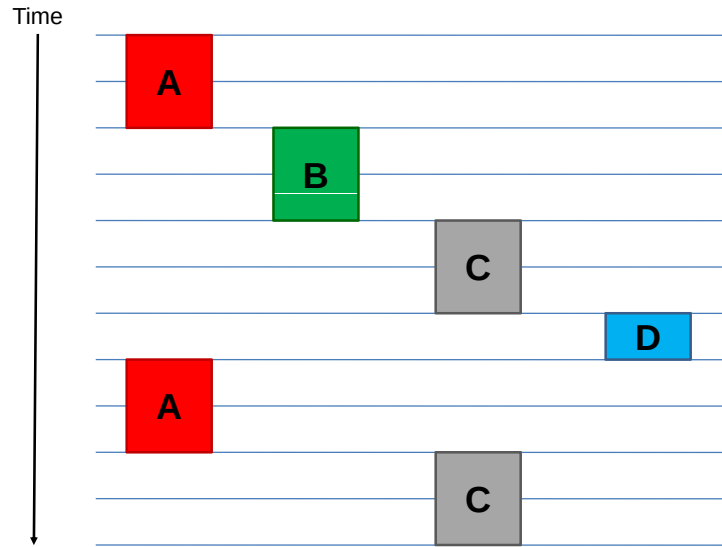
Processes

- Definition: "an instance of a program in execution"
- A process provides a context for the executing program: current memory state, open files, register contents, PC, environment variables
- When you type in the name of a program in the shell, the shell creates a new process in which that program will run

Multitasking

- Even though it appears to us that many programs run at the same time, in reality, only one process is able to use a CPU at a time
- The CPU is divided between all the currently running processes by the OS, which forces processes to be suspended
- To each process, it appears as if the process has sole control of the CPU, and logical flow of one process isn't affected by others (except for IPC)

Multitasking example



User and kernel modes

- Some instructions are restricted (or *privileged*), so that only the OS can execute them
 - e.g., halting the CPU, performing I/O, creating processes
- When a process needs to do one of these things, the process must enter kernel mode
- Normally, processes are in user mode

User and kernel modes, cont.

- When in user mode, trying to perform a privileged instruction or access kernel-reserved memory results in a fatal protection fault
- Should a process need to do one of these things, it needs to use the system call interface to access these instructions/areas of memory

Implementation of multitasking

- To move processes in and out of the CPU, the kernel needs to maintain the context (execution state) of each process
- Data structures used to maintain context:
 - page table, stores processes' address spaces
 - process table, stores information about each process
 - file table, lists which files are opened by each process

Context switching

- When a process is to be moved out of the CPU, the context is saved and the context of the incoming process is restored
- This is how the kernel schedules processes in the CPU
- Processes can be switched out whenever the kernel deems it necessary or appropriate
 - when a process is waiting for I/O
 - when a process has been using the CPU for a long time

System calls

- "The system call is the fundamental interface between an application and the Linux kernel."
-- the **syscalls** manual page
- System calls include functions to do things like
 - open/close/create/delete files
 - read from/write to files
 - create processes
 - read the system clock

Error handling

- When a system call fails, it sets the global integer **errno** to a specific number to indicate what went wrong, and often returns -1
- When you call any system call, you need to check to see if it failed
- For example, if the **fork()** function fails, it returns -1:

```
if ((pid = fork()) < 0) {
    perror("fork error");
    exit(EX_OSERR);
}
```
- The book defines extra functions (e.g. **Fork()**) that call the corresponding existing function but do error handling, as above, to save on code writing

The **err.h** functions

- The header file **<err.h>** provides access to prototypes of functions that combine error reporting and exits:
- **void err(int exit_code, const char *fmt, ...);**
 - equivalent to:

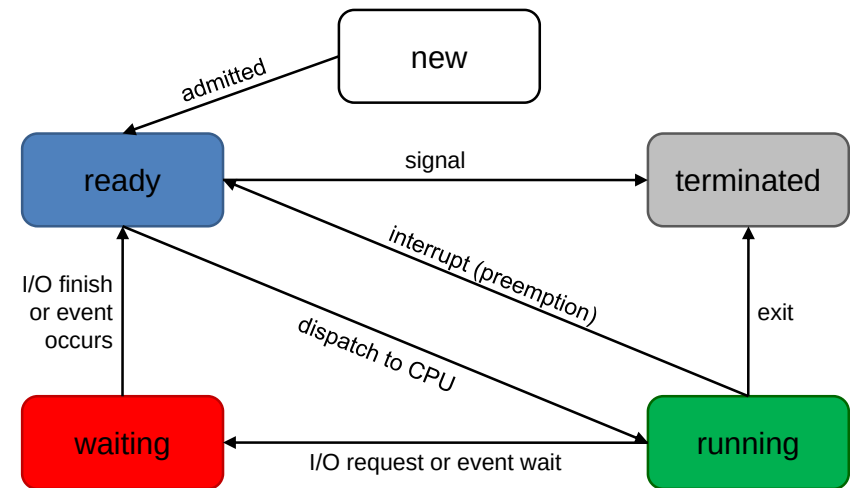
```
fprintf(stderr, fmt, ...);
fprintf(stderr, ": %s\n", strerror(errno));
exit(exit_code);
```
- **void errx(int exit_code, const char *fmt, ...);**
 - equivalent to:

```
fprintf(stderr, fmt, ...);
exit(exit_code);
```
- Non-exiting functions **warn()** and **warnx()** also exist
- Information on these can be found via "man **errx**"
- Can be used to reduce error-checking code while maintaining flexibility in handling

Process life cycle

- Every process goes through several states (the exact type/number vary by OS)
- Typical states:
 - new: process has just been created
 - running: process is executing on the CPU
 - ready: process is waiting to be run
 - waiting: process is waiting for an event (e.g., I/O, signal)
 - terminated: process is finished executing

Process state transitions



Process IDs

- Every active process has a unique process ID, which can be obtained via **getpid()**
- The process ID of the current process' parent (the process which created the current process) can be obtained via **getppid()**
- Function prototypes:

```
#include <unistd.h>
#include <sys/types.h>

pid_t getpid(void);
pid_t getppid(void);
```