ULM: Ulm Lecture Machine (Bottom-Up Approach)

Assembly Programming: Pattern Book


Code
Translation
Recipes
ABC Program
Assembler Code


```
@ <stdio.hdr>

fn main()
{
    local ch: int;

    while (true) {
        ch = getchar();
        if (ch == EOF) {
            break;
        }
        if (ch >= 'a' && ch <= 'z') {
            ch = ch - 'a' + 'A';
        }
        putchar(ch);
    }
}
```


```
@ <stdio.hdr>

fn main()
{
    local ch: int;

    while (true) {
        ch = getchar();
        if (ch == EOF) {
            break;
        }
        if (ch >= 'a' && ch <= 'z') {
            ch = ch - 'a' + 'A';
        }
        putchar(ch);
    }
}
```

```
    .equ    ch,             1
    .equ    EOF,            0xFF
loop:
    getc    %ch
    subq    EOF,            %ch,        %0
    je      done
    subq    'a',            %ch,        %0
    jb      putchar
    subq    'z',            %ch,        %0
    ja      putchar
    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```

For getting started:

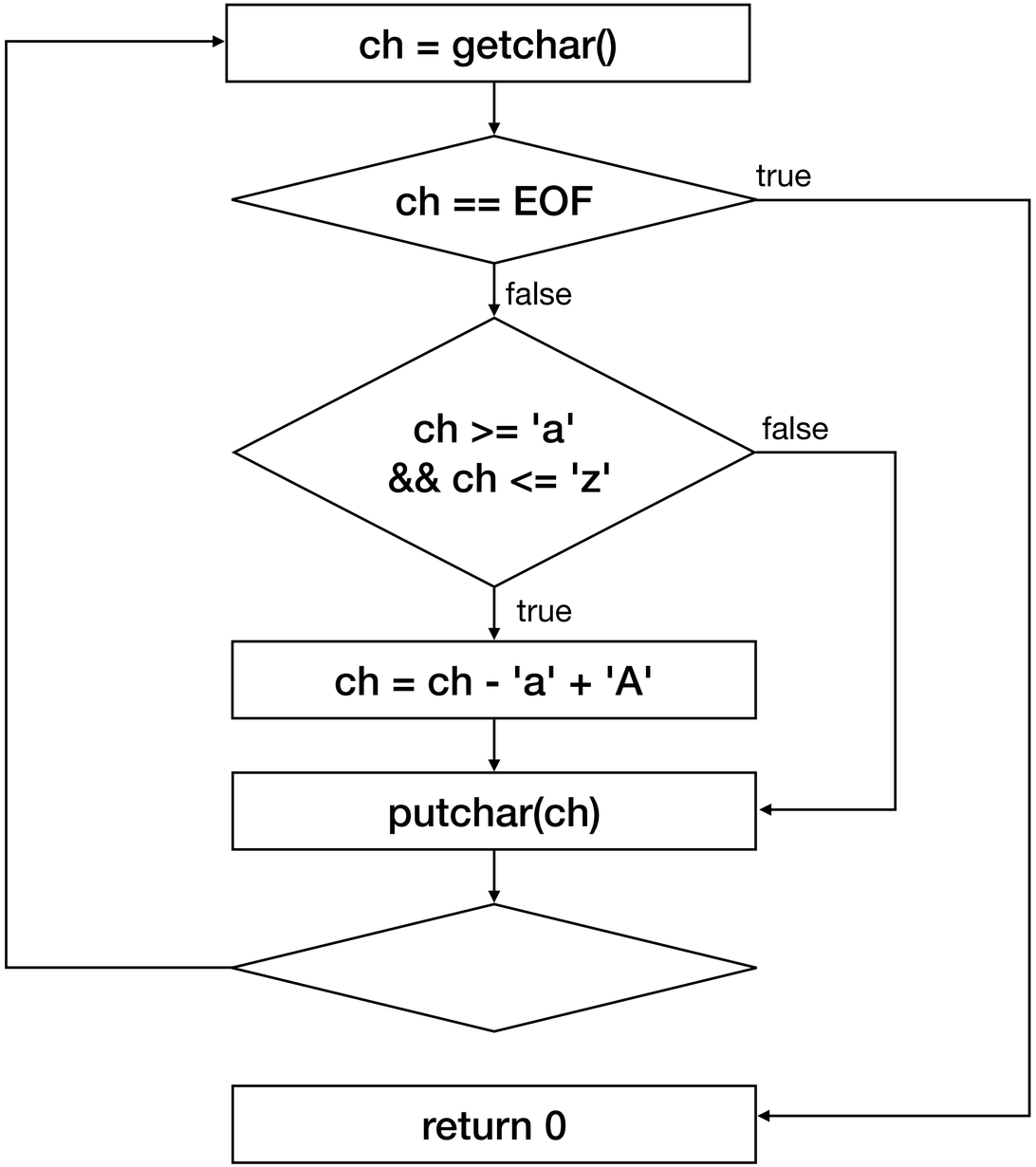
- No global variables
- assembly instructions for putchar, getchar
- Just one function: main
- Recall:
  - return value of main is the exit code
  - on Linux/macOS the exit code is truncated to 8 bits

```
@ <stdio.hdr>

fn main()
{
    local ch: int;

    while (true) {
        ch = getchar();
        if (ch == EOF) {
            break;
        }
        if (ch >= 'a' && ch <= 'z') {
            ch = ch - 'a' + 'A';
        }
        putchar(ch);
    }
}
```

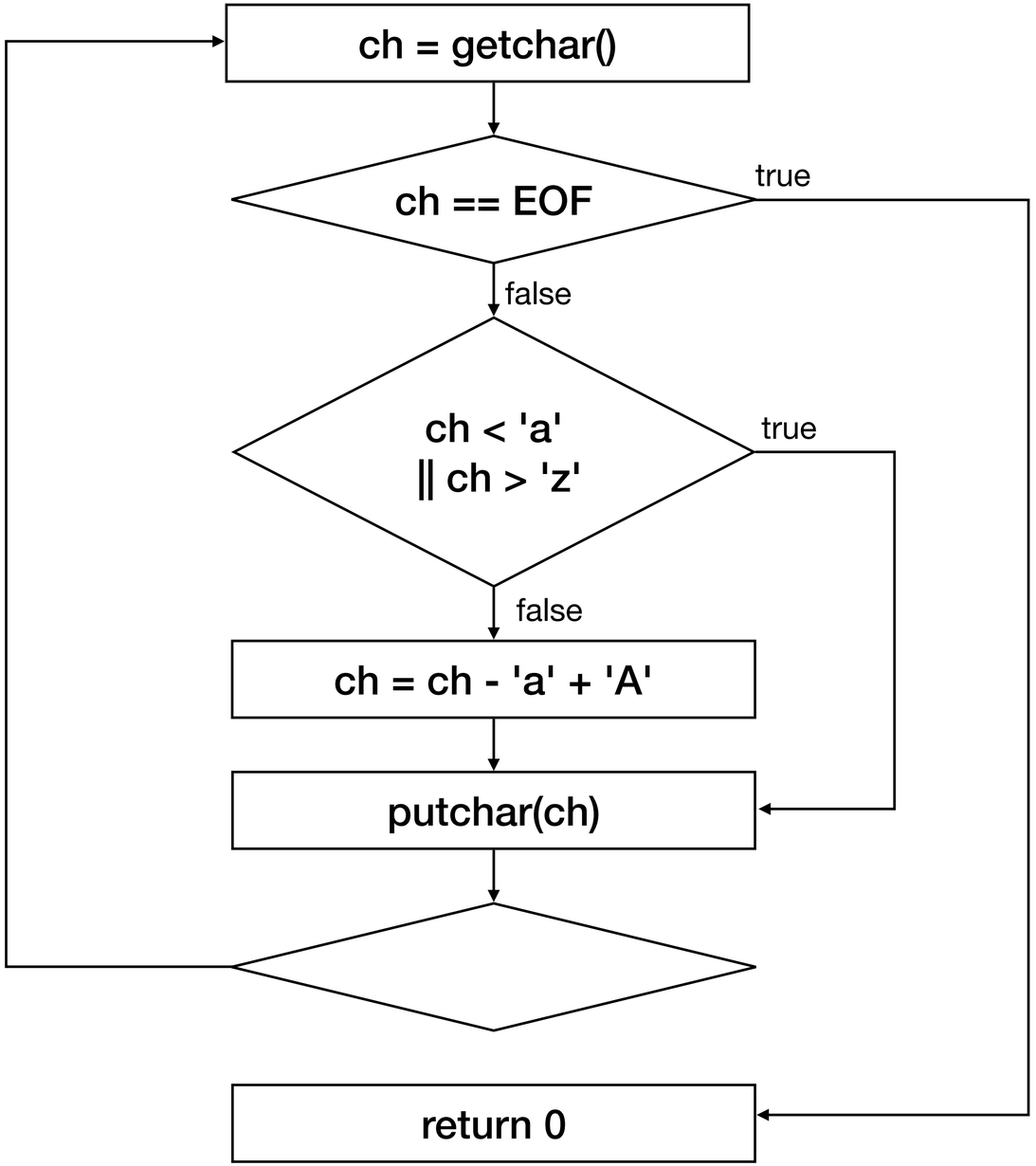
```
@ <stdio.hdr>

fn main(): int
{
    local ch: int;

    while (true) {
        ch = getchar();
        if (ch == EOF) {
            break;
        }
        if (ch >= 'a' && ch <= 'z') {
            ch = ch - 'a' + 'A';
        }
        putchar(ch);
    }

    return 0;
}
```

```
@ <stdio.hdr>

fn main(): int
{
    local ch: int;

label loop:
    ch = getchar();
    if (ch == EOF) {
        goto done;
    }
    if (ch >= 'a' && ch <= 'z') {
        ch = ch - 'a' + 'A';
    }
    putchar(ch);
    goto loop;

label done:
    return 0;
}
```

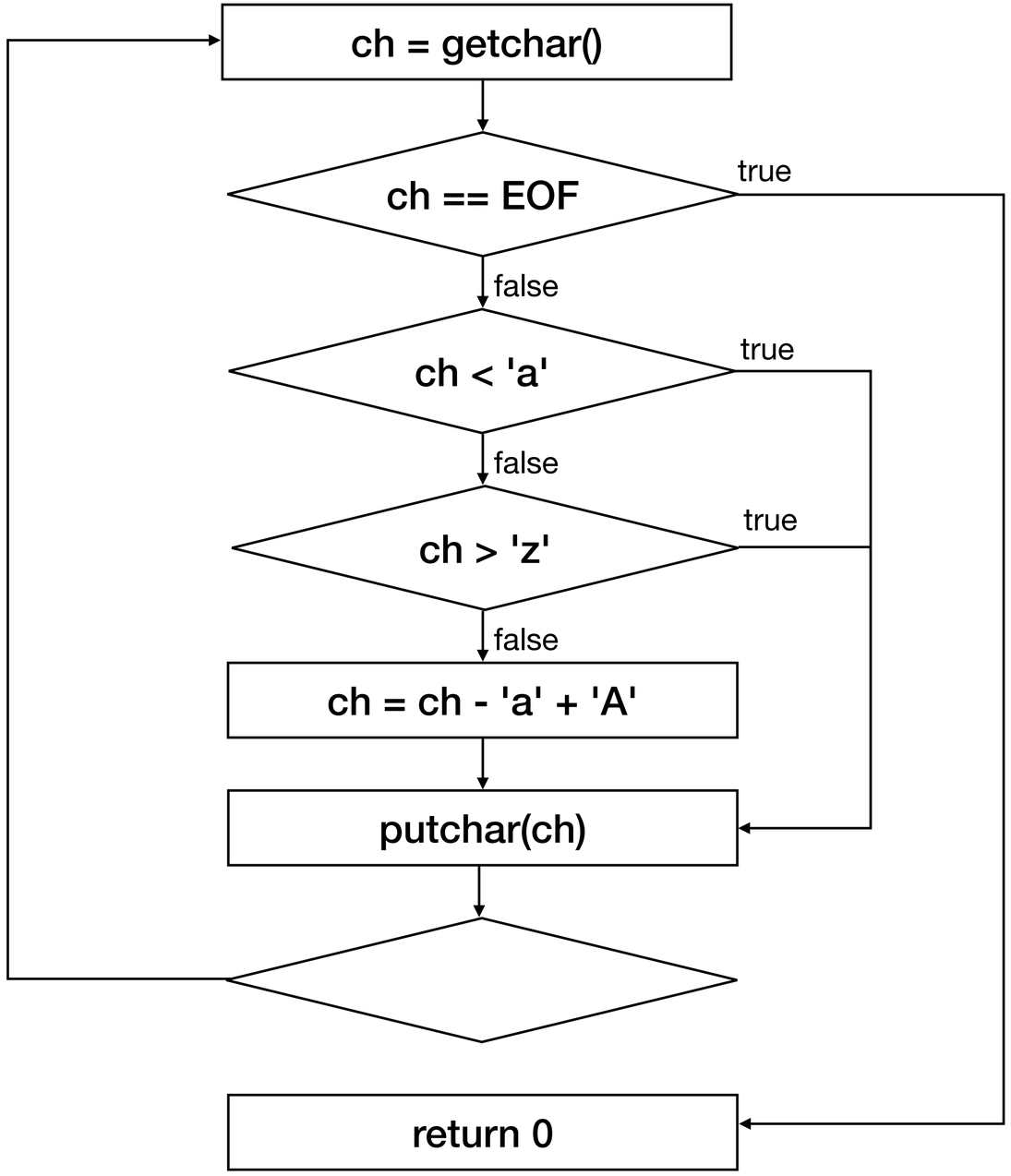
```
@ <stdio.hdr>

fn main(): int
{
    local ch: int;

label loop:
    ch = getchar();
    if (ch == EOF) {
        goto done;
    }
    if (ch >= 'a' && ch <= 'z') {
        ch = ch - 'a' + 'A';
    }
    putchar(ch);
    goto loop;

label done:
    return 0;
}
```



```
@ <stdio.hdr>

fn main(): int
{
    local ch: int;

label loop:
    ch = getchar();
    if (ch == EOF) {
        goto done;
    }
    if (ch < 'a' || ch > 'z') {
        goto putchar;
    }
    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
}
```



```
@ <stdio.hdr>

fn main(): int
{
    local ch: int;

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }
    if (ch < 'a') { goto putchar; }
    if (ch > 'z') { goto putchar; }
    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
}
```



```
@ <stdio.hdr>

fn main(): int
{
    local ch: int;

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }
    if (ch < 'a') { goto putchar; }
    if (ch > 'z') { goto putchar; }
    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
}
```

```c
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
loop:



putchar:

    jmp     loop

done:
    halt    0
```

```
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
loop:
    getc    %1

    subq    'a' - 'A',      %1,     %1
putchar:
    putc    %1
    jmp     loop

done:
    halt    0
```

```
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
    .equ    ch,             1

loop:
    getc    %ch



    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```

```
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
    .equ    ch,             1

loop:
    getc    %ch
    subq    0xFF,           %ch,        %0
    je      done



    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```

```
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
    .equ    ch,             1
    .equ    EOF,            0xFF
loop:
    getc    %ch
    subq    EOF,            %ch,        %0
    je      done



    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```

```
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
    .equ    ch,             1
    .equ    EOF,            0xFF
loop:
    getc    %ch
    subq    EOF,            %ch,        %0
    je      done
    subq    'a',            %ch,        %0
    jb      putchar


    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```

```
// ch is a signed integer

label loop:
    ch = getchar();
    if (ch == EOF) { goto done; }

    if (ch < 'a') { goto putchar; }

    if (ch > 'z') { goto putchar; }

    ch = ch - 'a' + 'A';
label putchar:
    putchar(ch);
    goto loop;

label done:
    return 0;
```

```
    .equ    ch,             1
    .equ    EOF,            0xFF
loop:
    getc    %ch
    subq    EOF,            %ch,        %0
    je      done
    subq    'a',            %ch,        %0
    jb      putchar
    subq    'z',            %ch,        %0
    ja      putchar
    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```

```
        .section
__TEXT,__text,regular,pure_instructions
        .build_version macos, 14, 0
        .globl  _main
_main:
LBB0_1:
        callq   _getchar
        cmpl    $-1, %eax
        je      LBB0_6
        cmpl    $97, %eax
        jb      LBB0_5
        cmpl    $122, %eax
        ja      LBB0_5
        addl    $-32, %eax
LBB0_5:
        movl    %eax, %edi
        callq   _putchar
        jmp     LBB0_1
LBB0_6:
        xorl    %eax, %eax
        retq
```

```
    .equ    ch,             1
    .equ    EOF,            0xFF
loop:
    getc    %ch
    subq    EOF,            %ch,        %0
    je      done
    subq    'a',            %ch,        %0
    jb      putchar
    subq    'z',            %ch,        %0
    ja      putchar
    subq    'a' - 'A',      %ch,        %ch
putchar:
    putc    %ch
    jmp     loop

done:
    halt    0
```