

Building a B compiler for the PDP-8

Robert Clausecker
[<fuz@fuz.su>](mailto:fuz@fuz.su)

The B Language



CC-BY-SA Peter Hamer, Wikimedia

```
#include <stdio.h>

int main(void) {
    int i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }

    return (0);
}
```

```
#include <stdio.h>

int main(void) {
    int i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }

    return (0);
}
```

```
int main(void) {
    extrn printf;
    int i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }

    return (0);
}
```

```
int main(void) {
    extrn printf;
    int i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }

    return (0);
}
```

```
main() {
    extrn printf;
    auto i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }

    return (0);
}
```

```
main() {  
    extrn printf;  
    auto i, f = 1;  
  
    for (i = 0; i < 10; i++) {  
        printf("%d! = %d\n", i, f);  
        f *= i;  
    }  
  
    return (0);  
}
```

```
main() {
    extrn printf;
    auto i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }
}
```

```
main() {
    extrn printf;
    auto i, f = 1;

    for (i = 0; i < 10; i++) {
        printf("%d! = %d\n", i, f);
        f *= i;
    }
}
```

```
main() {
    extrn printf;
    auto i = 0, f = 1;

    while (i < 10) {
        printf("%d! = %d\n", i, f);
        f *= i++;
    }
}
```

```
main() {
    extrn printf;
    auto i = 0, f = 1;

    while (i < 10) {
        printf("%d! = %d\n", i, f);
        f *= i++;
    }
}
```

```
main() {
    extrn printf;
    auto i 0, f 1;

    while (i < 10) {
        printf("%d! = %d*n", i, f);
        f *= i++;
    }
}
```

Historical Implementations



Threaded Code

name	code	PDP-7	description
autop	a	040000	push automatic variable
binop	b	100000	binary operator
consop	c	140000	push address sized constant
ifop	f	200000	jump if stack value is zero
etcop	n	240000	misc. function
setop	s	300000	set stack pointer
traop	t	340000	jump
unaop	u	400000	unary operator
extop	x	440000	push external variable
aryop	y	500000	define automatic vector
	z		switch statement

The PDP-8



- DEC minicomputer from 1965
- 12 bit architecture
- 4096 words of memory
- 8 instructions
- widely successful
- see it at this exhibition!

MAIN INSTRUCTIONS

OP	I	P	Y
0	1	2	3 4 5 6 7 8 9 10 11

P=1: $Y \leftarrow Y + \text{CURRENT PAGE}$

I=1: $Y \leftarrow C(Y)$

0 AND	$AC \leftarrow AC \wedge C(Y)$	AND
1 TAD	$L:AC \leftarrow AC + C(Y)$	TWO'S COMP. ADD
2 ISZ	$C(Y) \leftarrow C(Y)+1.$ $C(Y) = 0:$ SKIP	INDEX, SKIP IF ZERO
3 DCA	$C(Y) \leftarrow AC.$ $AC \leftarrow 0$	DEPOSIT, CLEAR AC
4 JMS	$C(Y) \leftarrow PC.$ $PC \leftarrow Y+1$	JUMP TO SUBROUTINE
5 JMP	$PC \leftarrow Y$	JUMP
6 IOT		IO TRANSFER
7 OPR		OPERATE

OPERATE GROUP 1

1	1	1	0	CLA	CLL	CMA	CML	RAR	RAL	rot2	IAC
0	1	2	3	4	5	6	7	8	9	10	11

7200 CLA	$AC \leftarrow 0$	CLEAR AC
7100 CLL	$L \leftarrow 0$	CLEAR L
7040 CMA	$AC \leftarrow \sim AC$	COMPLEMENT AC
7020 CML	$L \leftarrow \sim L$	COMPLEMENT L
7010 RAR	$L:AC \leftarrow \text{rotate } L:AC \text{ right}$	ROTATE AC RIGHT
7004 RAL	$L:AC \leftarrow \text{rotate } L:AC \text{ left}$	ROTATE AC LEFT
7012 RTR	RAR twice	ROTATE TWICE RIGHT
7006 RTL	RAL twice	ROTATE TWICE LEFT
7001 IAC	$L:AC \leftarrow AC + 1$	INDEX AC
7041 CIA	$L:AC \leftarrow \sim AC + 1$	COMPLEMENT, INDEX AC

OPERATE GROUP 2

1	1	1	1	CLA	SMA	SZA	SNL	SKP	OSR	HLT	0
0	1	2	3	4	5	6	7	8	9	10	11

7600 CLA	AC \leftarrow 0	CLEAR AC
7500 SMA	AC < 0: SKIP	OR
7440 SZA	AC = 0: SKIP	
7420 SNL	L \neq 0: SKIP	
7510 SPA	AC \geq 0: SKIP	AND
7450 SNA	AC \neq 0: SKIP	
7430 SZL	L = 0: SKIP	
7410 SKP	reverse SKIP condition	SKIP
7404 OSR	AC \leftarrow AC v SR	OR SWITCH REGISTER
7402 HLT	HALT	HALT