

Язык Си: стандарты, основные концепции. Исполнение программы.

Стандарты языка Си

Начало 70-х: появление языка Си

1978: Kernighan, Ritchie (K&R)

1989: ANSI C (C89)

1999: C99

2011: C11



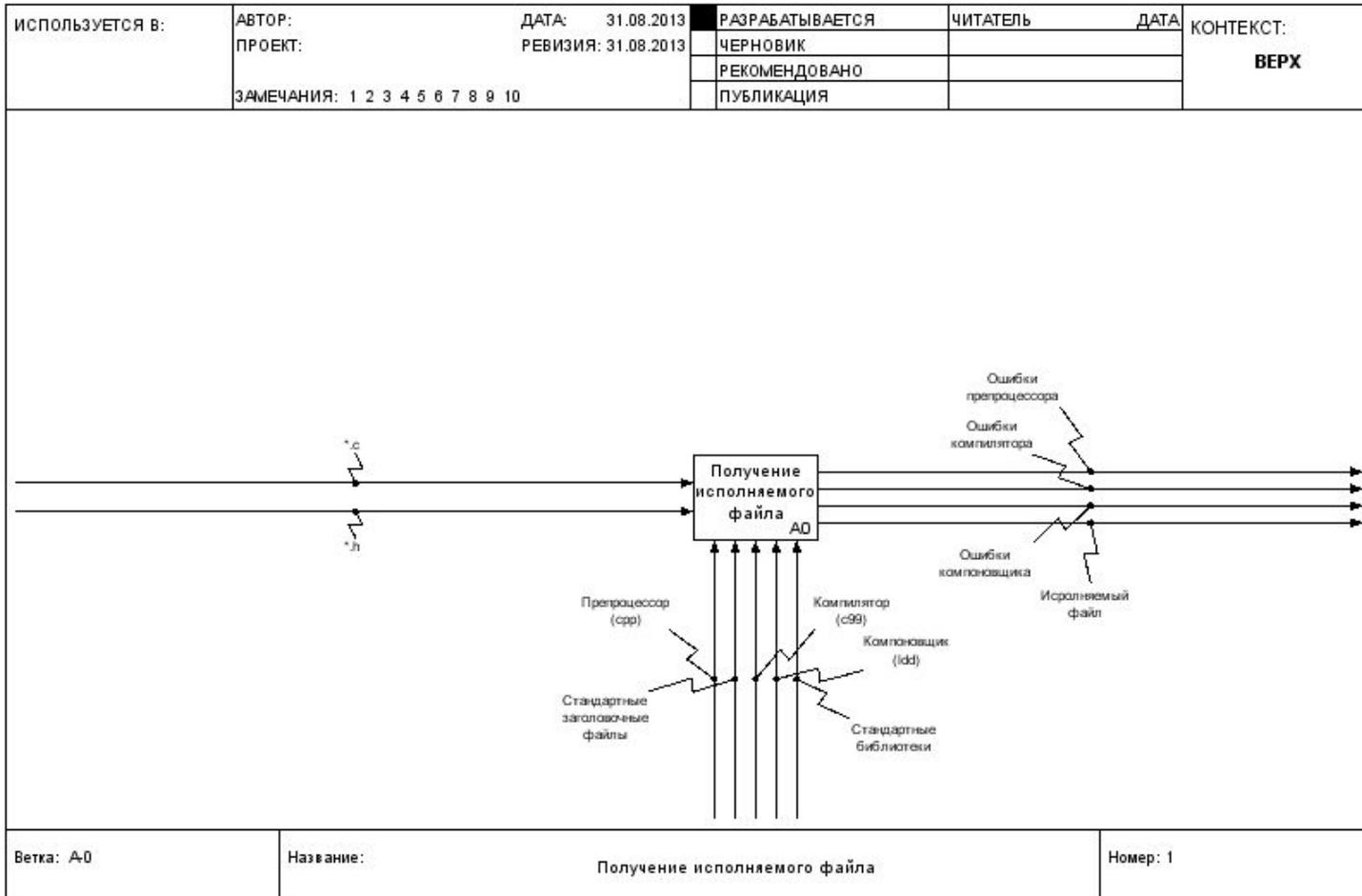
Основные концепции языка Си

- Си - язык "низкого" уровня
- Си - "маленький" язык с однопроходным компилятором
- Си предполагает, что программист знает, что делает

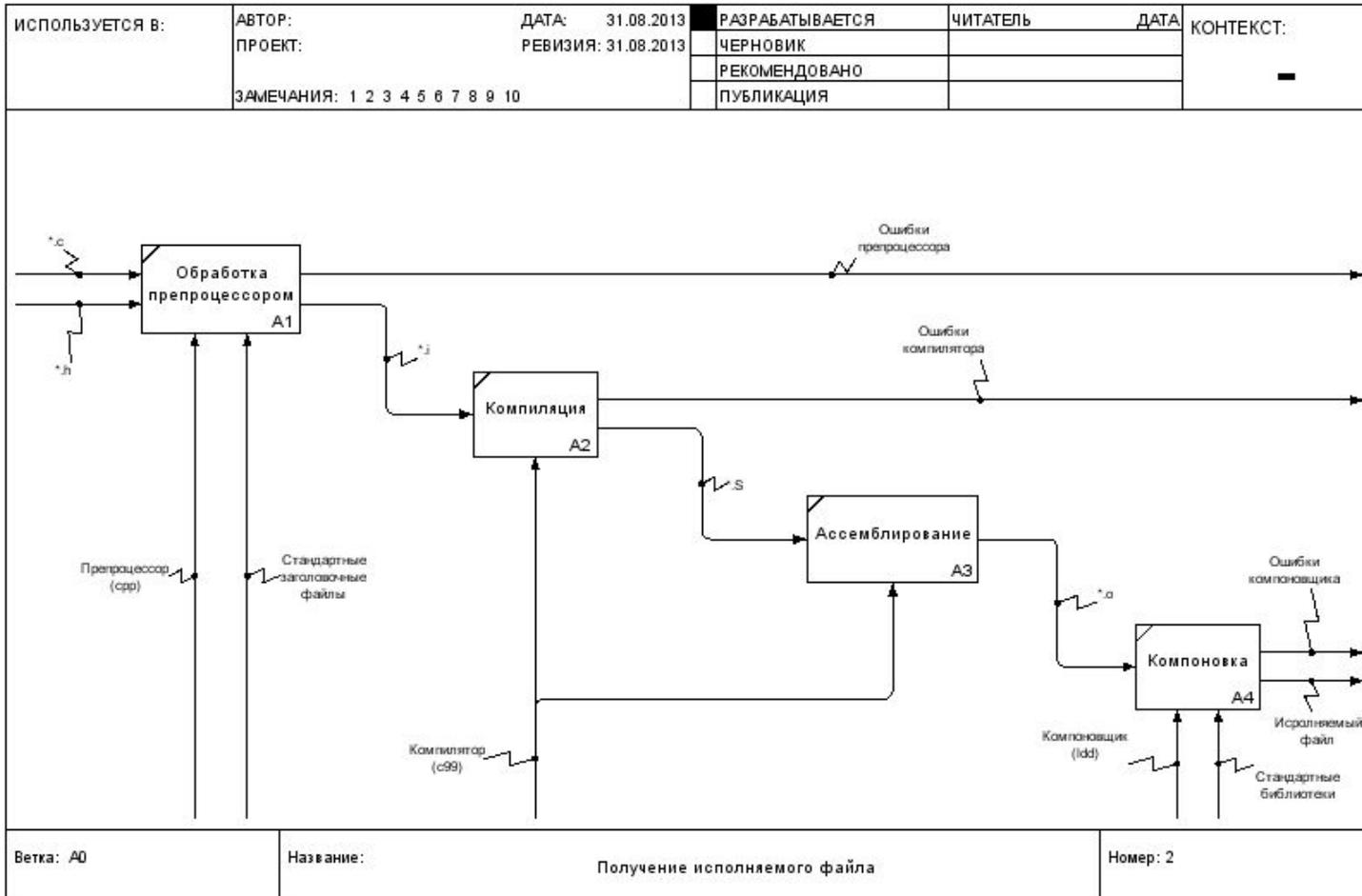
Получение исполняемого файла

```
0. // hello.c
1. #include <stdio.h>
2.
3. #define N 3
4.
5. int main(void)
6. {
7.     // Вывод фразы N раз
8.     for(int i = 0; i < N; i++)
9.         puts("Hello, world!\n");
10.
11.     return 0;
12. }
```

Получение исполняемого файла



Получение исполняемого файла



A1: обработка препроцессором

```
сpp -o hello.i hello.c (или сpp hello.c > hello.i)
```

hello.c - 165 байт, hello.i - 17226 байт

A2: трансляция на язык ассемблера

```
с99 -S -mams=intel hello.i
```

hello.c - 165 байт, hello.s - 677 байт

Результат работы препроцессора

```
...  
  
int __attribute__((__cdecl__)) __attribute__((__nothrow__)) puts  
    (const char*);  
  
...  
  
int main(void)  
{  
  
    for(int i = 0; i < 3; i++)  
        puts("Hello, world!\n");  
  
    return 0;  
}
```

Результат работы компилятора

```
.file "hello.c"
.intel_syntax noprefix
.section .rodata
.LC0:
.string "Hello, world!\n"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
push ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
mov ebp, esp
.cfi_def_cfa_register 5
and esp, -16
sub esp, 32
mov DWORD PTR [esp+28], 0
jmp .L2
```

Результат работы компилятора

```
.L3:
    mov  DWORD PTR [esp], OFFSET FLAT:.LC0
    call puts
    add  DWORD PTR [esp+28], 1
.L2:
    cmp  DWORD PTR [esp+28], 2
    jle  .L3
    mov  eax, 0
    leave
    .cfi_restore 5
    .cfi_def_cfa 4, 4
    ret
    .cfi_endproc
.LFE0:
    .size  main, .-main
    .ident  "GCC: (SUSE Linux) 4.6.2"
    .section .comment.SUSE.OPTs,"MS",@progbits,1
    .string "ospwg"
    .section .note.GNU-stack,"",@progbits
```

A3: ассемблирование в объектный файл

```
c99 -c hello.s
```

```
hello.c - 165 байт, hello.o - 1120 байт
```

A4: компоновка

```
ld -o hello.exe hello.o
```

```
hello.c - 165 байт, hello.exe - 4894 байт
```

ОПЦИИ КОМПИЛЯТОРА И КОМПОНОВЩИКА

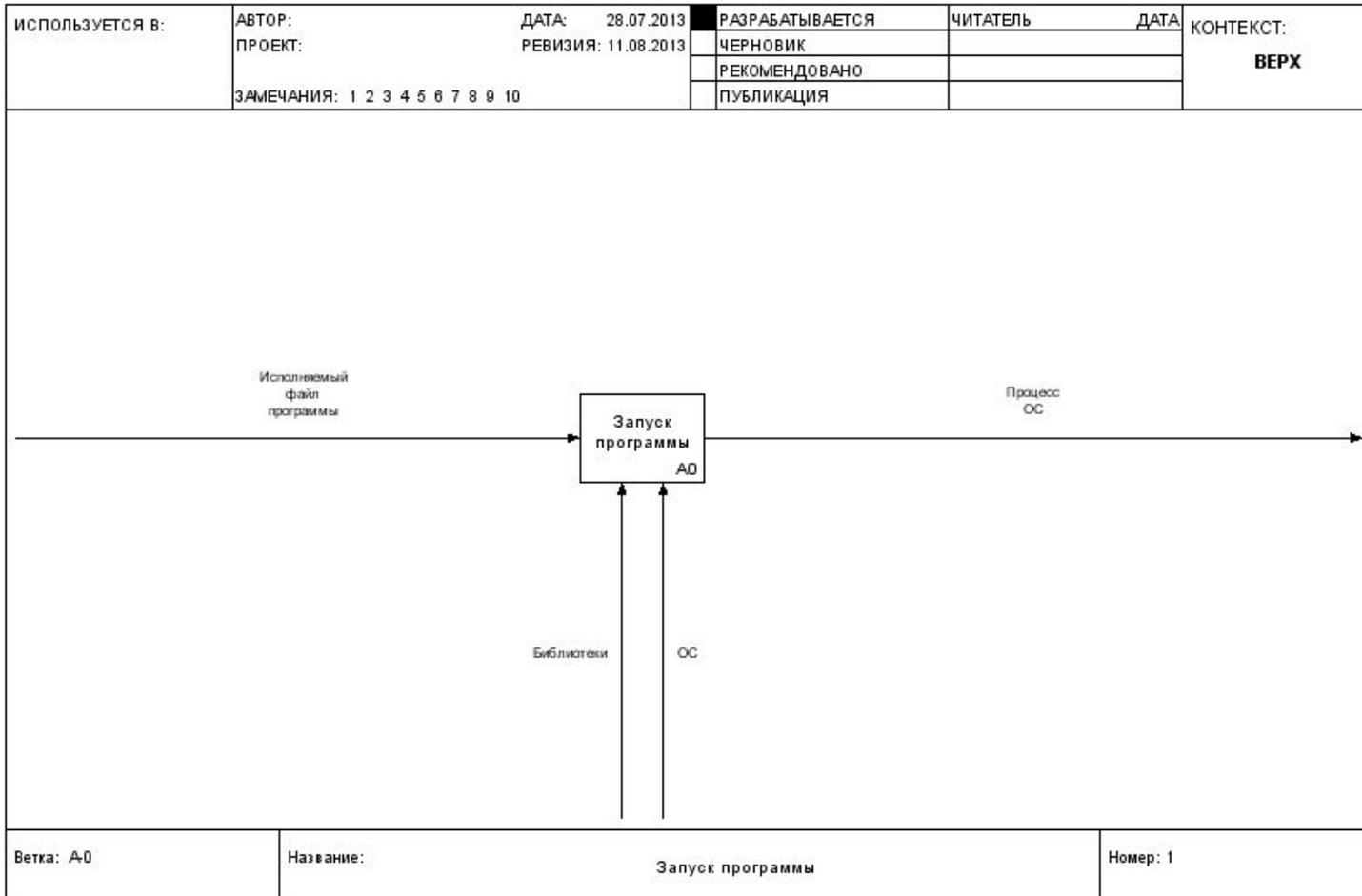
с99 [опции] [выходной_файл] файл_1 [файл_2]

- -pedantic
- -Wall
- -Werror
- -c (--compile)
- -o <ИМЯ>
- -g[level] (--debug)

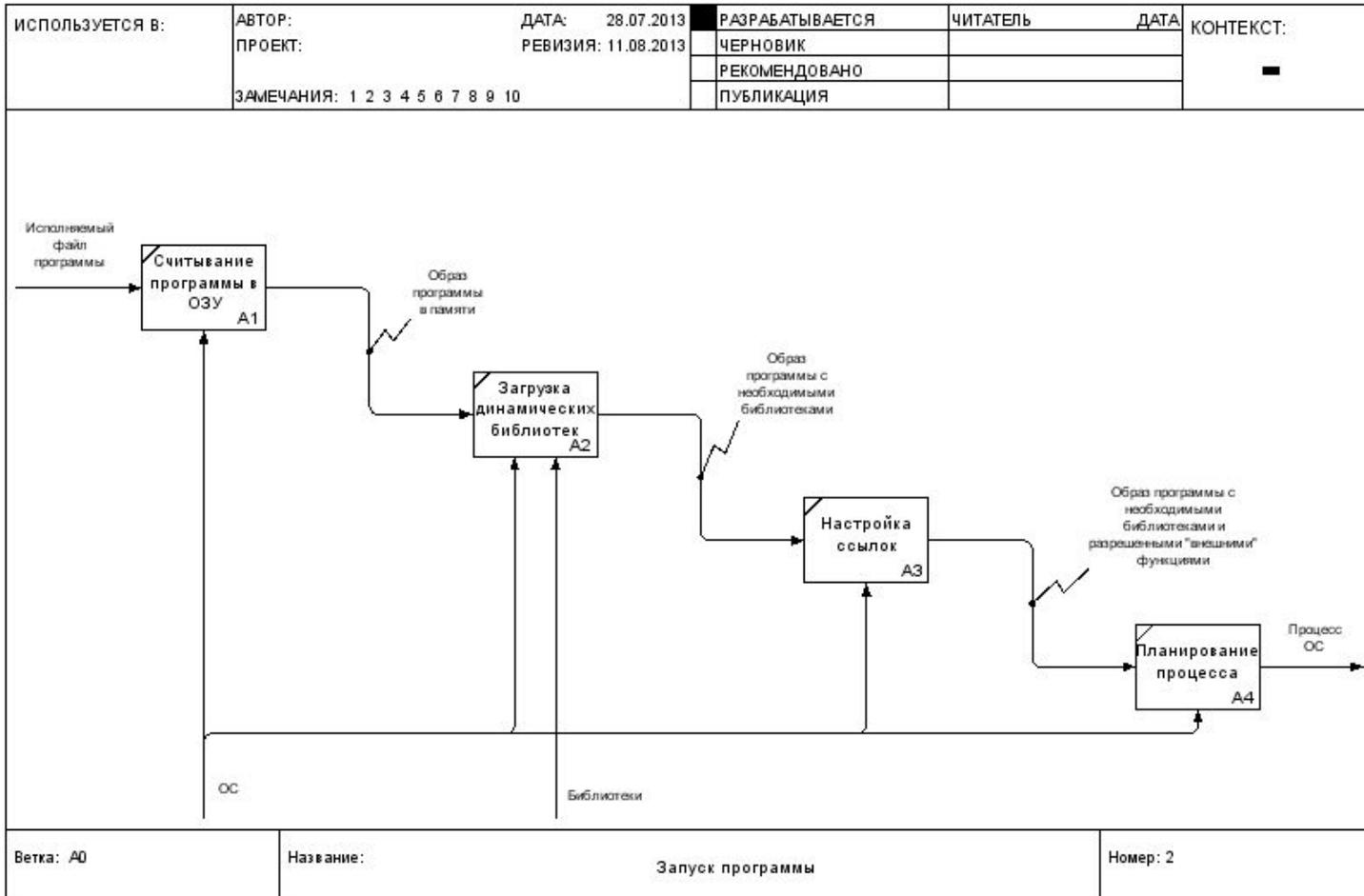
Представление о формате исполняемого файла

| |
|-----------------|
| Заголовок 1 |
| ... |
| Заголовок N |
| Секция text |
| Секция bss |
| Секция data |
| Секция rodata |
| Таблица импорта |
| ... |

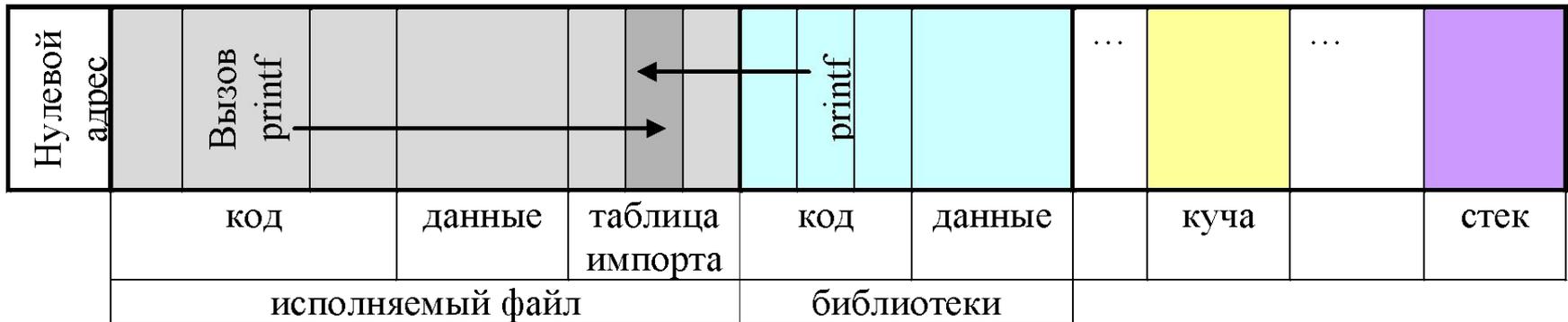
Запуск программы



Запуск программы



Абстрактная память и процесс.



Функция main

```
int main(void) ;  
int main(int, char** argv) ;
```

Значение, возвращаемое main

```
@echo off  
hello.exe  
if errorlevel 1 goto err  
if errorlevel 0 goto ok  
goto fin  
:err  
echo ERROR!  
goto fin  
:ok  
echo OK  
:fin
```

Параметры функции main

```
#include <stdio.h>

int main(int argc, char** argv)
{
    for(int i = 0; i < argc; i++)
        printf("%s\n", argv[i]);

    return 0;
}
```

