

ДИСЦИПЛИНА

Операционные системы

(полное наименование дисциплины без сокращений)

ИНСТИТУТ

Институт информационных технологий

КАФЕДРА

информационных технологий в атомной энергетике

(полное наименование кафедры)

ВИД УЧЕБНОГО

Лекция

МАТЕРИАЛА

(в соответствии с пп 1-11)

ПРЕПОДАВАТЕЛЬ

Пугачев Андрей Васильевич

(фамилия, имя, отчество)

СЕМЕСТР

IV семестр 2024 – 2025 учебный год

(указать семестр обучения, учебный год)

Тема № 1: ”Введение”

«Операционные системы»

МИРЭА – Российский технологический университет

Москва. 2024-2025 у.г.

Пугачев Андрей Васильевич

тел. +7 (906) 777-67-47

e-mail: pugachev_a@mirea.ru

Тот военачальник искусен в
нападении,
чей **противник не знает**, что
ему защищать;
и тот искусен в обороне,
чей противник не знает, на
что ему напасть.

Сунь-Цзы, V в. до н.э.

Простая программа

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    printf("Hello , World!\n");
    return 0;
}
```

Трассировка “Hello, World” в Windows

1146	7:41:40.689 PM	1	KERNEL32.DLL	...RtlSetLastWin32Error (ERROR_SUCCESS)
1147	7:41:40.689 PM	1	hello_world.exe	printf ("Hello, world!" , ...)
1148	7:41:40.689 PM	1	MSVCR100D.dll	...EnterCriticalSection (0x00000000676fcdc8)
1149	7:41:40.689 PM	1	MSVCR100D.dll	...EnterCriticalSection (0x00000000676fcc38)
1150	7:41:40.689 PM	1	MSVCR100D.dll	...HeapAlloc (0x0000000000ff0000, 0, 4148)
1151	7:41:40.689 PM	1	MSVCR100D.dll	...LeaveCriticalSection (0x00000000676fcc38)
1152	7:41:40.692 PM	1	MSVCR100D.dll	...GetLastError ()
1153	7:41:40.692 PM	1	MSVCR100D.dll	...FlsGetValue (4)
1154	7:41:40.692 PM	1	MSVCR100D.dll	...SetLastError (ERROR_SUCCESS)
1155	7:41:40.692 PM	1	KERNEL32.DLL	...RtlSetLastWin32Error (ERROR_SUCCESS)
1156	7:41:40.692 PM	1	MSVCR100D.dll	...EnterCriticalSection (0x0000000000ff0c18)
1157	7:41:40.692 PM	1	MSVCR100D.dll	...GetLastError ()
1158	7:41:40.692 PM	1	MSVCR100D.dll	...FlsGetValue (4)
1159	7:41:40.692 PM	1	MSVCR100D.dll	...SetLastError (ERROR_SUCCESS)
1160	7:41:40.693 PM	1	KERNEL32.DLL	...RtlSetLastWin32Error (ERROR_SUCCESS)
1161	7:41:40.693 PM	1	MSVCR100D.dll	...GetConsoleMode (0x0000000000000020, 0x0000000000e9bc44)
1162	7:41:40.693 PM	1	KERNELBASE.dll	...NtDeviceIoControlFile (0x0000000000000010, NULL, NULL, NULL,
1163	7:41:40.693 PM	1	MSVCR100D.dll	...WriteFile (0x0000000000000020, 0x0000000000e9bc80, 15, 0x0000...
1164	7:41:40.693 PM	1	KERNELBASE.dll	...NTWriteFile (0x0000000000000020, NULL, NULL, NULL, 0x0000...
1165	7:41:40.693 PM	1	MSVCR100D.dll	...LeaveCriticalSection (0x0000000000ff0c18)
1166	7:41:40.693 PM	1	MSVCR100D.dll	...LeaveCriticalSection (0x00000000676fcdc8)

Трассировка “Hello, World” в Linux

```
1: execve("/tmp/a.out", ["/tmp/a.out"], [/* 43 vars */]) = 0
2: brk(0) = 0x602000
3: mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb2e31e000
4: access("/etc/ld.so.preload", R_OK) = -1 EMOENT (No such file or directory)
5: open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
6: fstat(3, {st_mode=S_IFREG|0644, st_size=181057, ...}) = 0
7: mmap(NULL, 181057, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fbb2e2f1000
8: close(3) = 0
9: open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
10: read(3, "177ELF21100000000030>0100000M200000"... , 832) = 832
11: fstat(3, {st_mode=S_IFREG|0755, st_size=1753440, ...}) = 0
12: mmap(NULL, 3865656, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fbb2dd4e000
13: mprotect(0x7fbb2dd4e000, 2097152, PROT_NONE) = 0
14: mmap(0x7fbb2e0f4000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a6000) = 0x7fbb2e0f4000
15: mmap(0x7fbb2e0fa000, 15416, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fbb2e0fa000
16: close(3) = 0
17: mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb2e2f0000
18: mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb2e2ef000
19: mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb2e2ee000
20: arch_prctl(ARCH_SET_FS, 0x7fbb2e2ef700) = 0
21: mprotect(0x7fbb2e0f4000, 16384, PROT_READ) = 0
22: mprotect(0x600000, 4096, PROT_READ) = 0
23: mprotect(0x7fbb2e31f000, 4096, PROT_READ) = 0
24: munmap(0x7fbb2e2f1000, 181057) = 0
25: fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 3), ...}) = 0
26: mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb2e31d000
27: write(1, "Hello, World\n", 14) = 14
28: exit_group(0) = ?
29: +++ exited with 0 +++
```

Трассировка “Hello, World”

[illegible]

1146	741-40-630.PSA	1	KERNEL32.DLL	SetLastError(DWORD) (ERROR_SUCCESS)
1147	741-40-630.PSA	1	kernel_base.dll	print ("Host, user")
1148	741-40-630.PSA	1	MSVCRT100.DLL	EnterCriticalSection (0x0000000000000000f) (true)
1149	741-40-630.PSA	1	MSVCRT100.DLL	EnterCriticalSection (0x0000000000000000f) (true)
1150	741-40-630.PSA	1	kernel_base.dll	HeapLock (0x0000000000000000, 0, 1148)
1151	741-40-630.PSA	1	MSVCRT100.DLL	LeaveCriticalSection (0x0000000000000000f) (true)
1152	741-40-630.PSA	1	MSVCRT100.DLL	GetLastError ()
1153	741-40-630.PSA	1	MSVCRT100.DLL	PhiGetView (4)
1154	741-40-630.PSA	1	MSVCRT100.DLL	SetLastError (ERROR_SUCCESS)
1155	741-40-630.PSA	1	KERNEL32.DLL	SetLastError (ERROR_SUCCESS)
1156	741-40-630.PSA	1	MSVCRT100.DLL	EnterCriticalSection (0x0000000000000000f) (true)
1157	741-40-630.PSA	1	MSVCRT100.DLL	GetLastError ()
1158	741-40-630.PSA	1	MSVCRT100.DLL	PhiGetView (4)
1159	741-40-630.PSA	1	MSVCRT100.DLL	SetLastError (ERROR_SUCCESS)
1160	741-40-630.PSA	1	KERNEL32.DLL	SetLastError (ERROR_SUCCESS)
1161	741-40-630.PSA	1	MSVCRT100.DLL	GetConsoleMode (0x00000000000000000002, 0x0000000000000000f) (4)
1162	741-40-630.PSA	1	MSVCRT100.DLL	GetDeviceControl (0x00000000000000000000, 0x0000000000000000f, NULL, NULL, 1161)
1163	741-40-630.PSA	1	MSVCRT100.DLL	WriteFile (0x00000000000000000002, 0x0000000000000000f, 15, 0x0000000000000000, NULL, NULL, 0x0000000000000000)
1164	741-40-630.PSA	1	MSVCRT100.DLL	SetLastError (ERROR_SUCCESS)
1165	741-40-630.PSA	1	MSVCRT100.DLL	LeaveCriticalSection (0x0000000000000000f) (true)
1166	741-40-630.PSA	1	MSVCRT100.DLL	EnterCriticalSection (0x0000000000000000f) (true)

Рекомендуемая литература



Пугачев А.В., Дружинин К.Ю., Зязин В.П.
Введение в операционные системы. Часть I.
(учебно-методическое пособие)



Х.М. Дейтел, П.Дж.Дейтел, Д.Р.Чофнес.
”Операционные системы”



Таненбаум Э.
”Современные операционные системы”

История развития ОС

1. “Open shop”.
2. Системы пакетной обработки (Batch Processing).
3. Многозадачные системы (Multiprogramming).
4. Системы с разделением времени (Timesharing).
5. Параллельное программирование (Concurrent Programming).
6. Персональный компьютер (Personal Computing).
7. Распределенные системы (Distributed Systems).

Определение

Операционная система – набор программ (программных модулей), позволяющих пользователю вычислительной системы решать собственные задачи и (или) производить запуск и обеспечивать работоспособность пользовательского программного обеспечения (пользовательских программных модулей)

Классификация ОС

1. По назначению.
2. По способу взаимодействия с пользователем.
3. По поддерживаемой аппаратной архитектуре.
4. По уровню переносимости.

По назначению

- ▶ ОС общего назначения;
- ▶ ОС реального времени;
- ▶ специальные ОС.

По способу взаимодействия с пользователем

- ▶ пакетные ОС;
- ▶ интерактивные ОС:
 - ▶ с диалоговым режимом (CUI);
 - ▶ с графическим режимом (GUI).
- ▶ встроенные ОС.

По поддерживаемой аппаратной архитектуре

- ▶ однопроцессорные;
- ▶ многопроцессорные;
- ▶ распределенные.

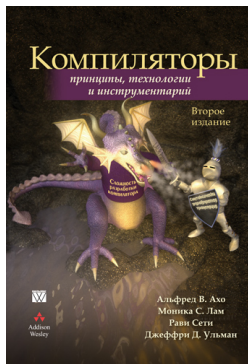
По уровню переносимости

- ▶ переносимые;
- ▶ не переносимые.

Способы сравнения ОС

- ▶ надежность;
- ▶ эффективность;
- ▶ удобство;
- ▶ масштабируемость;
- ▶ способность к развитию;
- ▶ переносимость;
- ▶ совместимость;
- ▶ стоимость использования.

Классы программных модулей



Ахо, Лам, Сети, Ульман.
Компиляторы: принципы, технологии
и инструменты

Классификация программных модулей

По возможности исполнения

- ▶ исполняемые;
- ▶ неисполняемые;
- ▶ гибридные.

По способу выполнения

- ▶ непосредственно исполняемые;
- ▶ транслируемые;
- ▶ интерпретируемые.

Взаимосвязь модулей

Типы связей программных модулей

- ▶ связь по данным;
- ▶ связь по коду.

Методы организации связей

- ▶ статические;
- ▶ динамические:
 - ▶ с использованием сторонних программ;
 - ▶ по средствам запроса.

Один код = две программы

Исходный код

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

```
int main(int argc, char* argv[])
{
    printf("Hello, □World!\n");
    return 0;
}
```

Варианты программ

[illegible]

1147	7-41-40.609 PM	1	KERNEL32.DLL	↳RtlSetWin32Error (ERROR_SUCCESS)
1148	7-41-40.609 PM	1	halco_wio!wdev	point ("halco_wio!wdev", ...)
1148	7-41-40.609 PM	1	MSVCRT100.dll	↳EnterCriticalSection (0x0000000070f9cd1d)
1149	7-41-40.609 PM	1	MSVCRT100.dll	↳EnterCriticalSection (0x0000000070f9cd1b)
1150	7-41-40.609 PM	1	MSVCRT100.dll	↳HeapAlloc (0x0000000000000000, 0, 4148)
1151	7-41-40.609 PM	1	MSVCRT100.dll	↳LeaveCriticalSection (0x0000000070f9cd1b)
1152	7-41-40.609 PM	1	MSVCRT100.dll	↳GetLastError ()
1153	7-41-40.609 PM	1	MSVCRT100.dll	↳FltGetValue (4)
1154	7-41-40.609 PM	1	MSVCRT100.dll	↳SetLastError (ERROR_SUCCESS)
1155	7-41-40.609 PM	1	KERNEL32.DLL	↳RtlSetWin32Error (ERROR_SUCCESS)
1156	7-41-40.609 PM	1	MSVCRT100.dll	↳EnterCriticalSection (0x0000000000000010)
1157	7-41-40.609 PM	1	MSVCRT100.dll	↳GetLastError ()
1158	7-41-40.609 PM	1	MSVCRT100.dll	↳FltGetValue (4)
1159	7-41-40.609 PM	1	MSVCRT100.dll	↳SetLastError (ERROR_SUCCESS)
1160	7-41-40.609 PM	1	KERNEL32.DLL	↳RtlSetWin32Error (ERROR_SUCCESS)
1161	7-41-40.609 PM	1	MSVCRT100.dll	↳GetComModuleInfo (0x0000000000000002, 0x0000000000000004, 0x0000000000000000)
1162	7-41-40.609 PM	1	KERNELBASE.dll	↳HrInitDeviceControl (0x0000000000000001, NUL, NUL, NUL, 0x00000000)
1163	7-41-40.609 PM	1	MSVCRT100.dll	↳WriteFile (0x0000000000000002, 0x0000000000000000, 0x0000000000000000, 0x0000000000000000, 0x0000000000000000)
1164	7-41-40.609 PM	1	KERNELBASE.dll	↳WriteFile (0x0000000000000002, NUL, NUL, NUL, 0x00000000)
1165	7-41-40.609 PM	1	MSVCRT100.dll	↳LeaveCriticalSection (0x0000000000000010)
1166	7-41-40.609 PM	1	MSVCRT100.dll	↳LeaveCriticalSection (0x0000000070f9cd1d)

Библиотеки

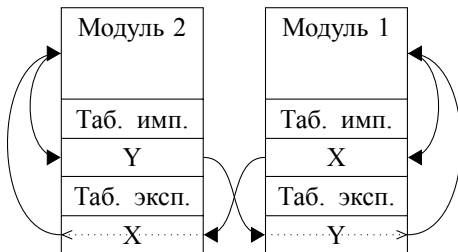
Типы библиотек

- ▶ статические;
- ▶ динамические.

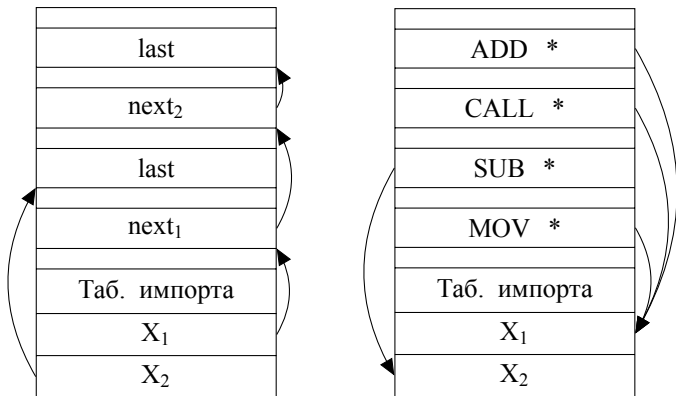
Процесс связывания программных модулей

Специальные таблицы

- ▶ экспортируемых символов;
- ▶ импортируемых символов.



Обработка ссылок к внешним символам

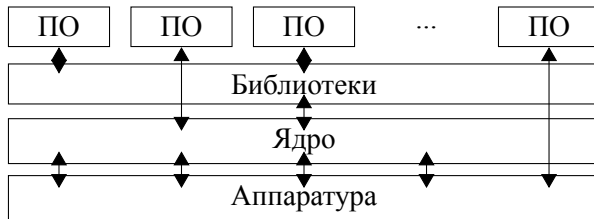


Общая структура операционной системы

Определение

Ядро — основной программный модуль, определяющий архитектуру всей ОС.

Общая архитектура ОС



Общая структура аппаратного обеспечения

1. Процессор.
2. Память:
 - ▶ оперативная;
 - ▶ долгосрочная.
3. Периферийные устройства.

Основные модули ОС

1. Менеджер процессов.
2. Менеджер памяти.
3. Файловые системы.
4. Подсистема ввода-вывода.

Типы ядер ОС

1. Монолитное.
2. Модульное.
3. Микроядро.
4. Экзоядро.

Вопросы?