

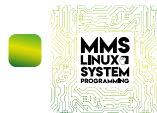
Системно програмиране за Линукс

Уводна лекция

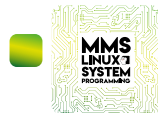
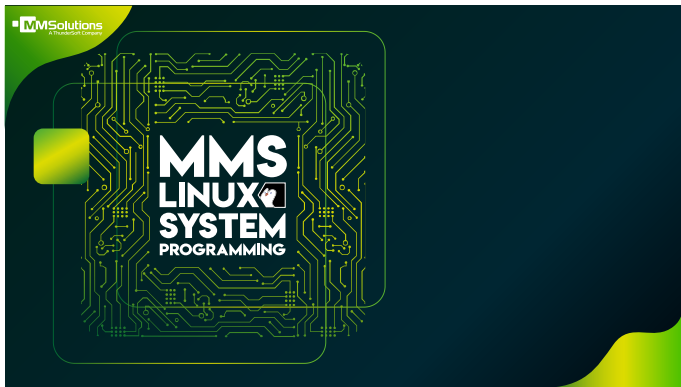
Ангел Чолаков



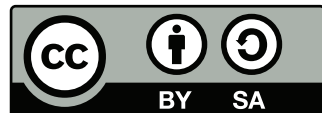
11.03.2021г.



За курса

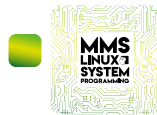


This work is licensed under a Creative Commons
“Attribution-ShareAlike 4.0 International” license.

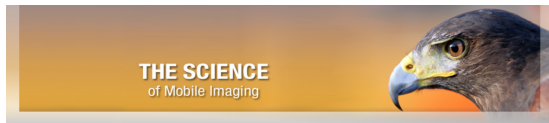


Съдържание I

- 1 Представяне
- 2 Подготовка
- 3 График и цели на курса
- 4 Въведение
- 5 Системно програмиране
- 6 Свободен софтуер
- 7 Движението GNU
- 8 Появата на Линукс ядрото
- 9 Вградените системи днес
- 10 Предимства и организация на Линукс ОС
- 11 Първи стъпки с Линукс
- 12 Заключение



MM Solutions - лидерът в мобилните камера решения



■ Области на експертиза:

- основана през 2001г., произлизаща от Daisy Group;
- алгоритми за обработка на изображения в мобилни телефони;
- управление и тюнинг на камера сензори;
- разработка на решения за Андроид базирани мобилни устройства;
- проектиране и прототипизиране на специализирани хардуерни платформи



Защо да посещавам този курс?

■ Запитвали ли сте се:

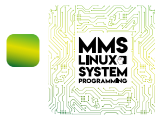
- какви принципи са в основата на съвременните ОС?
- как да се възползваме програмно от възможностите на ОС?
- как се управлява достъпът до системните ресурси?
- как се извършва синхронизацията на потребителски процеси?
- как да научим повече за Линукс ОС и Линукс ядрото?



Подготовка

За курса ще са необходими:

- най-вече ентузиазъм и мотивация;
- базови познания по С програмиране;
- компютър (може и стар преносим такъв);
- Интернет достъп и удобен текстови редактор;
- Не се изисква предварителен опит с Линукс



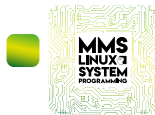
Планирано провеждане и организация

- планирано начало - 22.03.2021г.;
- продължителност - 10 или 11 учебни седмици;
- лекции - 3 учебни часа седмично;
- упражнения - 2 учебни часа седмично;
- споделяне на материалите - в <https://github.com/MM-Solutions>;
- практическа ориентация и текущи задачи;
- **записки?** - няма напълно все още, но пък има множество Интернет ресурси :)



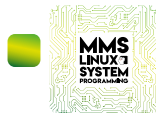
Схема на оценяване

- **либерално оценяване** - преминал или не;
- 30 % - от участие и проявен интерес;
- 70 % - от изпълнение на текущи задачи;



Основни цели на курса

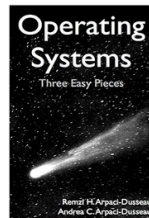
- стимулира интереса към създаване на Posix C проекти с GNU Make, GCC и GIT;
- да послужи като въведение в света на Линукс системното програмиране;
- разгледа основни понятия, свързани със структурата на една ОС;
- спомогне за последващо по-задълбочено изучаване на механизмите на Линукс ядрото



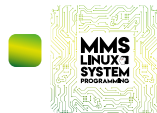
Препоръчвани свободно достъпни онлайн книги



автор: William Shotts



автори: R. H. и A. C Arpaci-Dusseau



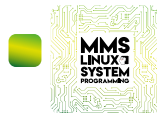
Как се разграничават системното и приложно програмиране?

■ Системно програмиране

- обезпечава функционирането на една компютърна система;
- създава среда за изпълнение на други приложения;
- подпомага работата на други софтуерни компоненти

■ Приложно програмиране

- реализира функция, ориентирана към потребителя;
- възползва се от наличните системни компоненти



Микропроцесорите са навсякъде днес...



Кратка предистория

Еволюцията на микропроцесорните архитектури

- през 1971г. се появява първият микроконтролер - Intel 4004;
- стимулира се появата на ново поколение микропроцесорни системи;
- довежда до революция в електронната индустрия

”Busicom Calculator с Intel 4004”

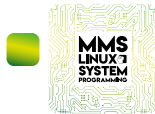


pic. by leighklotz, CC BY 2.0 via creativecommons.org



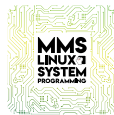
Развитие на системното програмиране

- **Появяват се платформени езици от високо ниво като:**
 - ALGOL и др.;
 - PL/1;
 - C;
- **Зараждат се първите комплексни операционни системи:**
 - Burroughs Master Control Program, 1961
 - Multics, 1969;
 - Unix - Bell labs, 1971;



Атрибути на системното програмиране

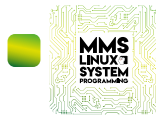
- разработчиците се възползват от възможностите на наличния хардуер оптимално;
- произвежданият програмен продукт е оптимизиран и адаптиран към средата и времевите изисквания на приложението;
- използват се специализирани инструменти за анализ, профилиране и дебъгване;
- подпомага обезпечаването на среда за изпълнение на програмни задания на високо ниво;



Какво представлява свободният софтуер и какво общо има със системното програмиране?

Смисълът не е еднозначен...

- терминът **”свободен софтуер”** се появява през 80-те години на 20-ти век от Richard Stallman;
- не трябва да считаме свободния софтуер като такъв, който непременно се разпространява безплатно;
- става дума за наблягане върху произтичащите **свободи и отговорности**, а не задължително липса на цена.



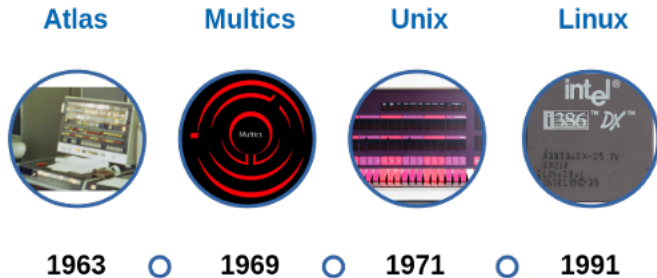
Движението GNU



pic. by OsamaK - CC BY-SA 2.0 via creativecommons.org



Появата на Линукс ядрото



Времева линия на ОС, вдъхновили създаването на Линукс ядрото



Появата на Линукс ядрото



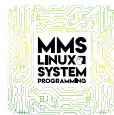
pic. by krd CC-BY-SA-4.0 via
Wikimedia Commons

- Финландецът Linus Torvalds се вдъхновява силно от идеологията на Richard Stallman и GNU движението;
- недоволен от лицензните ограничения на Unix-подобната ОС Minix, Torvalds решава да разработи свободно лицензирана алтернатива;
- версия 1.0 на Линукс ядрото е оповестена през 1994 г.;



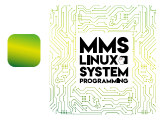
Основни характеристики на вградените системи днес

- специализирано предназначение;
- използване на микропроцесорни решения (вместо микроконтролери);
- наличие на широка гама комуникационни интерфейси - I2C, SPI, USB, Ethernet и др.;
- възможност за обработка на данни в реално време;
- използване на богат набор периферни компоненти

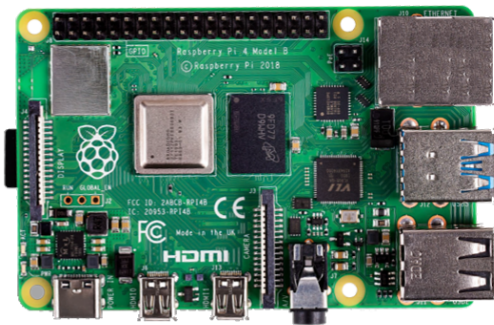


Предизвикателства и цели в разработката на вградени системи

- нарастваща паралелизация на изчисленията;
- въвеждане на новаторски потребителски интерфейси;
- адаптиране на модела на софтуерните приложения за многозадачна работа;
- обработка на все по-голям обем поточни данни в реално време;
- конструиране на нов тип функционални блокове



Популярността на Линукс в контекста на вградените системи

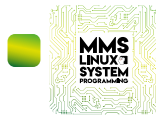


Raspberry PI - най-популярната Линукс хоби развойна среда за обучителни цели



Предимства на Линукс базираните решения

- отворен характер на разработка;
- утвърдени практики на сътрудничество;
- устойчива структура и богата системна функционалност;
- нарастваща общност от разработчици и ползватели;
- по-ниска себестойност на операционната система в сравнение със съществуващи алтернативи;



Организация на една Линукс ОС

■ В основата е Линукс ядрото

- обезпечава необходимата системна функционалност;
- служи като среда за изпълнение и управление на потребителски програми;
- налице са множество разклонения на изходния код според версията и степента на стабилност;
- колекцията от системни библиотеки и помощни програми, събрани редом с ядрото, се нарича **дистрибуция**;

■ Моделът на разработка е децентрализиран

- дейностите по координиране са поемат от различни утвърдени организации;
- отделните дистрибуции се различават според избраната стратегия за обновяване и достъпност на пакетите



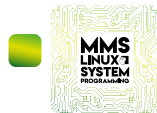
Страница на "The Linux Kernel Archives" портала

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release

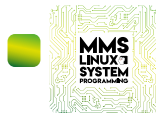
5.11.2

mainline:	5.12-rc1	2021-03-01	[tarball]	[patch]	[view diff]	[browse]			
stable:	5.11.2	2021-02-26	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.19	2021-02-26	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.101	2021-02-26	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.177	2021-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.222	2021-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.258	2021-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.258	2021-02-23	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20210226	2021-02-26					[browse]		



Основни подразделения в разработката на Линукс ядрото

- prepatch - издания, включващи последни нововъведения;
- mainline - издания, представляващи основното дърво на разработка;
- stable - стабилни издания, които произлизат от mainline линията на разработка;
- longterm - по-стари версии на ядрото, при които се прилагат само критични обновления;



Дотук добре, а как да започна?

- **Направете зареждаща USB флаш памет с Убунту, чрез например:**
 - <https://ubuntu.com/tutorials/create-a-usb-stick-on-windows>;
 - <https://releases.ubuntu.com/18.04.5/>;
 - необходимо е РС с поне 2 GB RAM и 4 GB USB флаш памет като минимум
- **Инсталирайте Линукс дистрибуция във виртуална машина:**
 - <https://askubuntu.com/questions/142549/how-to-install-ubuntu-on-virtualbox>;
 - <https://www.virtualbox.org/wiki/Downloads>;
 - подходящо решение, ако разполагате с по-мощна РС конфигурация



Примерни текстови редактори

Visual Studio Code



pic. by Microsoft, Public domain, via Wikimedia Commons

VIM

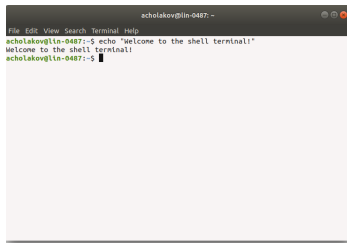


pic. by Fabián Alexis, CC BY-SA 3.0 via Wikimedia Commons

- <https://opensource.com/article/19/3/getting-started-vim>
- <https://code.visualstudio.com/docs/introvideos/basics>



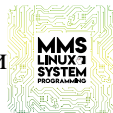
С какво ще се забавляваме по-натам?



```
acholakov@lin-0487: ~  
File Edit View Search Terminal Help  
acholakov@lin-0487:~$ echo "Welcome to the shell terminal!"  
Welcome to the shell terminal!  
acholakov@lin-0487:~$
```

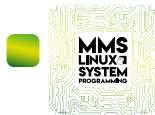
Примерно изображение на команден терминал

- Ще взаимодействаме през shell - текстово ориентиран команден интерпретатор;
- Предстои да научим основни команди за работа с файловата система и създаване на С приложения



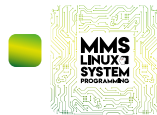
Основни команди за инсталиране и управление на пакети чрез apt-get

- apt-get update - обновяване на метаданните;
- apt-get install <pkg >- инсталиране на пакет;
- apt-get upgrade - обновяване на инсталираните пакети без премахване на съществуващи;
- apt-get dist-upgrade - обновяване на инсталираните пакети и зависимости, но с риск от премахване на пакети;
- apt-get remove <pkg >- премахване на пакет заедно с неговите конфигурационни файлове;
- apt-get autoremove - премахване на пакети, които вече не са необходими;
- apt-get purge <pkg >- цялостно премахване на пакет заедно с конфигурационните файлове;
- apt-get clean - изчистване на хранилището с локално свалени пакети;
- apt-cache search <reg exp >- търсене на пакет със съвпадаща сигнатура;



Бележки по материалите и изложението

- материалът е изготвен с образователна цел;
- съставителите не носят отговорност относно употребата и евентуални последствия;
- съставителите се стремят да използват публично достъпни източници на информация и разчитат на достоверността и статута на прилаганите или реферирани материали;
- текстът може да съдържа наименования на корпорации, продукти и/или графични изображения (изобразяващи продукти), които може да са търговска марка или предмет на авторско право - ексклузивна собственост на съотнесените лица;
- референциите могат да бъдат обект на други лицензи и лицензни ограничения;
- съставителите не претендират за пълнота, определено ниво на качество и конкретна пригодност на изложението;
- съставителите не носят отговорност и за допуснати фактологически или други неточности;
- свободни сте да създавате и разпространявате копия съгласно посочения лиценз;



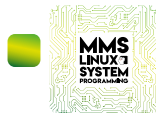
Референции към полезни източници на информация

- <https://en.wikipedia.org/>
- <https://www.gnu.org/>
- <https://www.kernel.org/>
- <https://tldp.org/LDP/intro-linux/html/index.html>
- https://en.wikipedia.org/wiki/Intel_4004
- <https://www.raspberrypi.org/>
- <https://linuxcommand.org/tlcl.php>
- <https://pages.cs.wisc.edu/~remzi/OSTEP/>
- <https://www.cs.cornell.edu/courses/cs2043/2019sp/schedule.html>
- https://en.wikipedia.org/wiki/Linux_kernel
- https://en.wikipedia.org/wiki/Operating_system
- <https://www.debian.org/doc/manuals/debian-reference/ch02.en.html>
- <https://search.creativecommons.org/>



Референции към графични материали

- <https://search.creativecommons.org/photos/cb69d45e-6190-46f7-be94-46c97da0d2e5>
- <https://search.creativecommons.org/photos/a9221794-1e84-4328-9dd2-22c817bb9f30>
- [https://en.wikipedia.org/wiki/Atlas_\(computer\)](https://en.wikipedia.org/wiki/Atlas_(computer))
- https://en.wikipedia.org/wiki/Intel_80386



Благодаря Ви за вниманието!

