

۷ کارنیل، بزرگترین شبکه موفقیت ایرانیان می باشد، که افرادی زیادی توانسته اند با آن به موفقیت برسند، فاطمه رتبه ۱۱ کنکور کارشناسی، محمد حسین رتبه ۶۸ کنکور کارشناسی، سپیده رتبه ۳ کنکور ارشد، مریم و همسرش راه اندازی تولیدی مانتو، امیر راه اندازی فروشگاه اینترنتی، کیوان پیوستن به تیم تراکتور سازی تبریز، میلاد پیوستن به تیم صبا، مهسا تحصیل در ایتالیا، و.... این موارد گوشه از افرادی بودند که با کارنیل به موفقیت رسیده اند، شما هم می توانید موفقیت خود را با کارنیل شروع کنید.

برای پیوستن به تیم کارنیلی های موفق روی لینک زیر کلیک کنید.

[www.karnil.com](http://www.karnil.com)

همچنین برای ورود به کانال تلگرام کارنیل روی لینک زیر کلیک کنید.

<https://telegram.me/karnil>

# ABC Book<sup>2013.1</sup>

نویسنده: ایمان امینی

نام کتاب: کتاب آرچ نسخه ۲۰۱۳.۱

نویسنده: ایمان امینی

ویرایش: علیرضا برهان

مجوز انتشار: GFDL

سال انتشار: اسفند ۱۳۹۱

## فهرست مطالب

۵.....	مقدمه نویسنده
۶.....	سپاس
۷.....	معرفی
۷.....	درباره این راهنما
۹.....	آماده‌سازی
۹.....	گرفتن آخرین رسانه (media) نصب
۹.....	بررسی صحت فایل دانلود شده
۱۰.....	رایت رسانه نصب
۱۶.....	راه‌اندازی رسانه نصب
۱۹.....	نصب
۱۹.....	تغییر زبان
۲۰.....	اتصال به اینترنت
۲۶.....	آماده کردن درایو ذخیره‌سازی
۴۷.....	متصل کردن پارتیشن‌ها
۴۸.....	انتخاب آینه
۴۹.....	نصب بیس سیستم
۵۰.....	ساخت fstab
۵۱.....	Chroot و پیکربندی پایه
۵۷.....	پیکربندی شبکه
۶۰.....	پیکربندی پک‌من
۶۱.....	ساخت initial ramdisk environment
۶۱.....	تنظیم پسورد ریشه و اضافه کردن کاربر معمولی
۶۲.....	نصب یک راه‌انداز (Boot Loader)
۶۷.....	پیاده کردن پارتیشن‌ها و شروع مجدد
۶۸.....	اضافات
۶۸.....	مدیر بسته (PACKage MANagement)
۷۹.....	مدیر سیستم (systemd)

- ۷۹..... بررسی وضعیت سیستم.
- ۸۲..... دیمون (DAEMON)
- ۸۳..... صدا
- ۸۶..... sudo
- ۸۷..... Yaourt
- ۸۹..... نصب X
- ۸۹..... نصب راه‌انداز ویدویی
- ۱۰۰..... نصب راه‌اندازهای ورودی
- ۱۰۰..... پیکربندی ایکس
- ۱۰۱..... تست X
- ۱۰۳..... فونت‌ها
- ۱۰۴..... انتخاب و نصب یک رابط گرافیکی
- ۱۱۵..... دانگرید برنامه (اسکرپیت دانگرید)
- ۱۱۷..... کار با chroot
- ۱۲۱..... معرفی تعدادی برنامه کاربردی و محبوب

# مقدمه نویسنده

هر روز نرم افزارهای آزاد جای خودشان را در زندگی ما بیشتر باز و ما را به استفاده از خود دعوت می‌کنند. مهم‌تر از قدرت برنامه‌ها و ابزارهای آزاد، فلسفه زیبا و احترام آن به کاربر، ما را به استفاده از آن‌ها ترغیب می‌کند. در این راهنما سعی شده که نحوه نصب یکی از قوی‌ترین توزیع‌های گنو/لینوکس یعنی آرچ لینوکس به زبان ساده توضیح داده شود تا شما خواننده عزیز به راحتی تجربه شیرینی را از گنو/لینوکس کسب کنید. فقط و فقط برای داشتن یک تجربه بی‌دردسر اکیداً توصیه می‌کنم که قسمت به قسمت این راهنما را با دقت هرچه تمام‌تر مطالعه نمائید و قبل از طرح سؤال و مشکل، موارد را بررسی کنید. سادگی آرچ لینوکس تمامی قسمت‌های این سیستم را برای شما قابل لمس می‌کند و شما بعد از نصب آن یک کاربر چشم و گوش بسته نیستید چرا که از نحوه عملکرد سیستم خود به کلی آگاه خواهید بود. این راهنما به صورت آزاد و تحت مجوز GFDL منتشر می‌شود بنابراین شما با توجه به مفاد این گواهی می‌توانید آن را به صورت آزاد دریافت کنید، به دیگران بدهید و بر اساس نیازهای روز و خودتان محتویاتش را تغییر دهید و منتشر کنید. برای دریافت نسخه قابل ویرایش کافیست به وبلاگ نویسنده<sup>۱</sup> مراجعه کنید. این شمائید که با کمک خود می‌توانید این راهنما را به‌روز و زنده نگه دارید.

با تشکر از همه دوستانی که در حین مواجه با مشکلات مختلف در طول نوشتن این کتاب، بنده را راهنمایی کردند، به ویژه دو دوست گرانقدر، علیرضا برهان و مهدی غفاری. امیدوارم این راهنما برای شما خواننده گرامی مفید واقع شود.

# سپاس

---

این قسمت به همراه امضاء، خطاب به دوستان است که این کتاب را از طریق وب سایت نویسنده به صورت اینترنتی خریداری نموده‌اند. برای مشاهده محتویات این صفحه بایستی از طریق صفحه مربوطه کتاب را خریداری نمائید.

<http://emanlog.com/?p=324>

خرید شما به ما اجازه ارائه کتاب‌های آزاد بیشتر و کارهایی از این دست را خواهد داد.

# معرفی

خوش آمدید. این سند شما را برای نصب و راه‌اندازی سیستم آرچ‌لینوکس با استفاده از اسکریپت‌های نصب راهنمایی می‌کند. آرچ‌لینوکس یک توزیع گنو/لینوکس ساده و سبک است که مناسب کاربران ماهر می‌باشد. این راهنما برای کاربران جدید آرچ تهیه شده اما حاوی اطلاعات پایه‌ای بسیار زیادی برای تمام کاربران است. قبل از نصب بهتر است قسمت سوالات و پاسخ‌های متداول<sup>۱</sup> را مشاهده نمایید.

## دربارهٔ این راهنما

این کتاب ترجمه‌ای از صفحهٔ راهنمای تازه‌کاران ویکی آرچ‌لینوکس به علاوهٔ تجربیات شخصی نویسنده در پاره‌ای از قسمت‌ها به همراه نکات اضافی است. ویکی آرچ‌لینوکس یک منبع بسیار عالی و بی‌نهایت جامع برای نصب و رفع مشکلات اولیه در آرچ محسوب می‌شود. کانال `(irc://irc.freenode.net/#archlinux)` و انجمن‌ها همچنین برای پاسخ به سؤالات و مشکلات حل نشده شما وجود دارد. قبل از طرح سوال مطمئن شوید که صفحهٔ `man` مربوط به هر دستور را برای راهنمایی بیشتر مطالعه نموده‌اید. اگر شما با صفحهٔ `man` ناآشنا هستید می‌توانید توسط صادر کردن دستور `man command` با آن بیشتر آشنا شوید.

اما برای شمایی که به زبان انگلیسی تسلط دارید آیا خواندن این راهنما مفید است؟ به صورت کلی توجه داشته باشید این راهنما نسبت به صفحهٔ راهنمای تازه‌واردان آرچ دارای خصوصیات است که در ادامه به تعدادی از آن‌ها اشاره خواهیم کرد:



- بزرگترین خصوصیت این راهنما زبان آن است که به پارسی نوشته شده است.
  - نویسنده این کتاب کاربر آرچ لینوکس بوده و در پاره‌ای از مواقع که توضیحات ویکی را ناکامل و یا نارس دیده اقدام به اضافه کردن توضیحات اضافی و لازم نموده است.
  - نصب و راه‌اندازی چند دسکتاپ و مدیرپنجره محبوب به صورت خلاصه توضیح داده شده و شما برای نصب آن‌ها نیازی به مطالعه صفحه ویکی مربوط به آن‌ها را نخواهید داشت.
  - آموزش نصب گلچین فونت‌های پارسی به علاوه معرفی یک سری از برنامه‌های پرکاربرد در آن گنجانده شده است.
  - نحوه نصب برنامه از AUR در این راهنما آموزش داده شده است.
  - نحوه دانگرید برنامه و Chroot زدن را فرا خواهید گرفت.
- در نهایت شما با ترجمه صفحه تازه‌واردان ویکی آرچ مواجه نیستید بلکه با گلچینی از آموزشات ویکی آرچ با اسکلت راهنمای تازه‌واردان طرفید.

تذکر: برای تجربه یک نصب موفق این راهنما را مرحله به مرحله و با دقت مطالعه نمائید. لطفاً تمام راهنما را بخوانید. قویاً توصیه می‌شود قبل از آنکه کاری انجام دهید هر بخش را کامل بخوانید.

## بخش اوّل

# آماده‌سازی

تذکر: اگر شما قصد نصب آرچ لینوکس را از یک توزیع دیگر گنو/لینوکس و یا یک دیسک زنده دارید لطفاً راهنمای آن را مطالعه نمائید. این راهنما همچنین می‌تواند شما را در نصب آرچ از طریق VNC یا SSH یاری کند.

## گرفتن آخرین رسانه (media) نصب

شما می‌توانید رسانه نصب رسمی آرچ را از صفحهٔ مربوطه دریافت کنید. آخرین نسخه موجود در این زمان 2012.12.01 می‌باشد.

## بررسی صحت فایل دانلود شده

با فرمان `cd` به مسیری که فایل دانلود شده را ذخیره کرده‌اید بروید و `sha1sum` را با روش زیر دریافت کنید.

```
$ sha1sum --check name_of_checksum_file.txt
```

بایستی شما "OK" را به معنی سالم بودن فایل دریافتی مشاهده کنید. (بقیه خطوط را نادیده بگیرید.) اگر پاسخ منفی بود بایستی فایل دانلود شده را دوباره دریافت کنید. md5sum هم عمل مشابه را جهت بررسی سلامت فایل انجام می‌دهد.

## رایت رسانه نصب

۱. ایمیج iso را بر روی CD یا DVD، با سخت‌افزار (رایتر) و نرم‌افزاری (برنامه رایت ایمیج مانند K3B) که ترجیح می‌دهید رایت کنید و به قسمت راه‌اندازی رسانه نصب بروید.

**تذکره:** بهتر است از لوح فشرده و درایو نوری خوب با سرعت پایین برای رایت ایمیج استفاده کنید. برخی از کاربران حتی از سرعت 2x یا 4x برای رایت ایمیج استفاده می‌کنند. اگر شما از CD رایت شده رفتار غیرعادی (عدم راه‌اندازی نصاب و مشکل در خواندن) مشاهده نمودید سعی بر رایت آن با حداقل سرعت پشتیبانی شده توسط سیستم خود نمائید.

۲. شما همچنین می‌توانید ایمیج را به یک حافظه قابل اتصال به پورت USB<sup>۱</sup> انتقال دهید.

### ۱. در گنو/لینوکس

**اخطار:** اطلاعات بر روی /dev/sdx به صورت غیرقابل بازگشت پاک خواهد شد. (دیوایس فلش شما را نشان می‌دهد و می‌تواند sda یا sdb یا ... باشد).

**نکته:** با استفاده از lsblk مطمئن شوید که دیوایس USB متصل نشده است و همچنین اطمینان حاصل کنید که از /dev/sdx به جای /dev/sdx1 استفاده می‌کنید (خطای بسیار رایج).

```
# dd bs=4M if=/path/to/archlinux.iso of=/dev/sdx
```

### ۲. در ویندوز

برنامه ایمیج رایتر را از صفحه لانچپد مربوطه<sup>۲</sup> دریافت و اجرا کنید. ایمیج و حافظه فلش را برای نوشتن اطلاعات انتخاب نمائید. توجه داشته باشید که این برنامه از شما فایل با پسوند img می‌خواهد و شما بایستی با تایپ \*.iso در باکس مربوط به Open ایمیج آرچ را به برنامه نشان دهید.

۱ "flash drive", "USB stick", "USB key" و غیره

۲ <http://launchpad.net/win32-image-writer>

همچنین می‌توانید با دریافت برنامه `Universal USB Installer` یک حافظه زنده لینوکس بسازید و از آن برای نصب آرچ بهره ببرید.

### ۳. در مک اواس ایکس

برای استفاده از `dd` در یک مک مراحل زیر را دنبال کنید. ابتدا حافظه فلش را به درگاه USB متصل کنید. سپس اواس-ایکس آن را به صورت خودکار متصل می‌کند در ادامه دستور

```
diskutils list
```

را در `Terminal.app` اجرا کنید. اکنون بایستی تشخیص دهید که دیوایس USB شما چه نام دارد. مال من `/dev/disk1` بود. حالا دستور زیر را برای پیاده کردن<sup>۲</sup> اجرا کنید:

```
diskutil unmountDisk /dev/disk1
```

اکنون با استفاده از دستور `dd` کار خود را پیش بگیرید:

```
dd if=image.iso of=/dev/disk1 bs=8192
```

```
20480+0 records in
20480+0 records out
167772160 bytes transferred in 220.016918 secs (762542 bytes/sec)
```

و در نهایت با دستور زیر قبل از خارج کردن فیزیکی فلش آن را کاملاً پیاده کنید:

```
diskutil eject /dev/disk1
```

## نصب از طریق شبکه

به جای رایت کردن رسانه نصب بر روی یک لوح فشرده یا حافظه شما می‌توانید از طریق شبکه ایمیج ایزو را راه‌اندازی نمایید. در این حالت شما بایستی یک سرور آماده داشته باشید.

ابتدا آخرین رسانه نصب رسمی را از آینه دلخواه<sup>۳</sup> خود دانلود کنید:

۱ <http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3>

۲ `umount`

۳ <https://wiki.archlinux.org/index.php/Mirrors>

```
$ wget http://hive.ist.unomaha.edu/archlinux/iso/latest/archlinux-2012.12.01-dual.iso
```

```
--2012-12-08 02:27:00-- http://hive.ist.unomaha.edu/archlinux/iso/latest/archlinux-2012.12.01-dual.iso
Resolving hive.ist.unomaha.edu (hive.ist.unomaha.edu)... 2620:d5:0:22bb::dead:beef, 137.48.187.208
Connecting to hive.ist.unomaha.edu (hive.ist.unomaha.edu)[2620:d5:0:22bb::dead:beef]:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 449839104 (429M) [application/octet-stream]
Saving to: 'archlinux-2012.12.01-dual.iso'

100%[=====] 449,839,104 110MB/s in 3.9s

2012-12-08 02:27:04 (110 MB/s) - 'archlinux-2012.12.01-dual.iso' saved [449839104/449839104]
```

در قدم بعدی ایمج را سوار کنید:

```
# mkdir -p /mnt/archiso
# mount -o loop,ro archlinux-2012.10.06-dual.iso /mnt/archiso
```

## تنظیمات سرور

اکنون شما به یک سرور TFTP و DHCP، برای پیکربندی شبکه، بارگزاری pxelinux/kernel/initramfs و در نهایت بارگزاری فایل سیستم ریشه نیاز دارید (به ترتیب).

## شبکه

NIC را بالا بیاورید و آدرس را به درستی تعیین کنید:

```
# ip link set eth0 up
# ip addr add 192.168.0.1/24 dev eth0
```

## DHCP + TFTP

شما به هر دوی DHCP و TFTP برای پیکربندی شبکه در مقصد نصب جهت تسهیل در امر رد و بدل کردن اطلاعات در بین سرور و کلاینت PXE نیاز دارید؛ dnsmasq هر دو کار را به آسانی انجام می‌دهد.

dnsmasq را نصب کنید:

```
# pacman -S dnsmasq
```

سپس پیکربندی dnsmasq :

```
# vim /etc/dnsmasq.conf
```

```
port=0
interface=eth0
bind-interfaces
dhcp-range=192.168.0.50,192.168.0.150,12h
dhcp-boot=/arch/boot/syslinux/pxelinux.0
dhcp-option-force=209,boot/syslinux/archiso.cfg
dhcp-option-force=210,/arch/
enable-tftp
tftp-root=/mnt/archiso
```

نوبت به اجرای dnsmasq رسیده است:

```
# systemctl start dnsmasq.service
```

## HTTP

خدا پدر تغییرات اخیر ایزوی آرچ لینوکس را بیامزد شما از این به بعد می‌توانید از طریق HTTP (archiso\_pxe\_http initcpio hook) یا NFS (archiso\_pxe\_nfs initcpio hook) راه‌اندازی را انجام دهید. در بین این همه جایگزین darkhttpd سبک‌ترین و کم اهمیت‌ترین مورد برای تنظیم است.

ابتدا darkhttpd را نصب کنید:

```
# pacman -S darkhttpd
```

سپس darkhttpd را با استفاده از /mnt/archiso به عنوان منبع ریشه اجرا کنید:

```
# darkhttpd /mnt/archiso
```

```
darkhttpd/1.8, copyright (c) 2003-2011 Emil Mikulic.
listening on: http://0.0.0.0:80/
```

## نصب

در این قسمت شما بایستی کشف کنید که چگونه از کلاینت بخواهید راه‌اندازی PXE در الویت قرار بگیرد. در حالت عادی، زمانی که سیستم را راه‌اندازی می‌کنید در گوشه پایین سمت چپ به شما نشان داده خواهد شد که کدام کلید را برای راه‌اندازی PXE با الویت بالاتر فشر دهید. به عنوان مثال در یک IBM x3650 کلید F12 منوی راه‌اندازی را نمایش می‌دهد.

## راه اندازی

با نگاهی بر `/var/log/messages.log` در سرور PXE شما می‌توانید ببینید که در مراحل اولیه راه‌اندازی چه اتفاقی افتاده است:

```
# grep dnsmasq /var/log/messages.log | cut -d ' ' -f5-
```

```
dnsmasq-dhcp[2544]: DHCPDISCOVER(eth1) 00:1a:64:6a:a2:4d
dnsmasq-dhcp[2544]: DHCP OFFER(eth1) 192.168.0.110 00:1a:64:6a:a2:4d
dnsmasq-dhcp[2544]: DHCPREQUEST(eth1) 192.168.0.110 00:1a:64:6a:a2:4d
dnsmasq-dhcp[2544]: DHCPACK(eth1) 192.168.0.110 00:1a:64:6a:a2:4d
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/pxelinux.0 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/whichsys.c32 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe_choose.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/ifcpu64.c32 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe_both_inc.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_head.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe32.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_pxe64.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso_tail.cfg to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/vesamenu.c32 to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/splash.png to 192.168.0.110
```

بعد از بارگذاری `archiso.cfg` و `pxelinux.0` از طریق TFTP شما با یک منوی راه‌اندازی `syslinux` به همراه چندین گزینه مواجه خواهید شد.

این گزینه:

```
Boot Arch Linux (x86_64) (HTTP)
```

یا این گزینه را بسته به معماری سیستم خود انتخاب کنید:

```
Boot Arch Linux (i686) (HTTP)
```

در ادامه داده‌های مربوطه (هسته و `initramfs`) از طریق TFTP رد و بدل می‌شوند:

```
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/x86_64/vmlinuz to 192.168.0.110
dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/x86_64/archiso.img to 192.168.0.110
```

اگر همه چیز خوب پیش برود شما باید فعالیت‌های بر روی `darkhttpd` که از `PXE-target` می‌آید را

مشاهده کنید؛ در اینجا هسته بایستی بر روی PXE-target و در init بارگزاری شود:

```
1348347586 192.168.0.110 "GET /arch/aitab" 200 678 "" "curl/7.27.0"
1348347587 192.168.0.110 "GET /arch/x86_64/root-image.fs.sfs" 200 107860206 "" "curl/7.27.0"
1348347588 192.168.0.110 "GET /arch/x86_64/usr-lib-modules.fs.sfs" 200 36819181 "" "curl/7.27.0"
1348347588 192.168.0.110 "GET /arch/any/usr-share.fs.sfs" 200 63693037 "" "curl/7.27.0"
```

بعد از این که فایل سیستم ریشه از طریق HTTP بارگیری شد شما وارد خط فرمان می‌شوید.

### بعد از راه‌اندازی

زمانی که فایل سیستم ریشه به طور کلی دانلود شد می‌توانید darkhttpd را بکشید و همچنین ایمج نصب را پیاده کنید:

```
# systemctl stop dnsmasq.service
# umount /mnt/archiso
```

### نصب بر روی یک ماشین مجازی

نصب بر روی یک ماشین مجازی راه خوبی برای آشنا شدن با آرچ لینوکس و پروسه نصب آن بدون خارج شدن از سیستم عامل فعلی و پارتیشن‌بندی ذخیره ساز اصلی (به عنوان مثال هارد دیسک شما) است. همچنین با این روش می‌توانید راهنما را در کنار خود داشته باشید. برخی از کاربران هم ترجیح می‌دهند یک سیستم آرچ لینوکس به صورت جداگانه و همیشه برای بررسی و آزمایش‌ها بر روی یک ماشین مجازی داشته باشند.

نمونه‌هایی از برنامه‌های شبیه‌ساز عبارتند از <sup>۱</sup>VirtualBox, <sup>۲</sup>VMware, <sup>۳</sup>QEMU, <sup>۴</sup>Xen, <sup>۵</sup>Varch و <sup>۶</sup>Parallels.

شیوه آماده‌سازی هر ماشین مجازی به نرم‌افزار آن بستگی دارد اما به صورت کلی شامل مراحل زیر می‌شوند:

- ۱ <https://wiki.archlinux.org/index.php/VMware>
- ۲ <https://wiki.archlinux.org/index.php/QEMU>
- ۳ <https://wiki.archlinux.org/index.php/Xen>
- ۴ <https://wiki.archlinux.org/index.php/Varch>
- ۵ <https://wiki.archlinux.org/index.php/VirtualBox>
- ۶ <https://wiki.archlinux.org/index.php/Parallels>



۱. ساخت دیسک مجازی که میزبان سیستم عامل میهمان می‌شود.
  ۲. پیکربندی مناسب پارامترهای ماشین مجازی.
  ۳. سوار کردن ایمیج iso. بر روی درایو CD ماشین مجازی.
  ۴. به قسمت راه‌اندازی رسانه نصب بروید.
- خواندن مقالات زیر برای شما در جهت استفاده بهینه از ماشین‌های مجازی می‌تواند مفید باشد:

- Arch Linux VirtualBox Guest <sup>۱</sup>
- Installing Arch Linux from VirtualBox <sup>۲</sup>
- VirtualBox Arch Linux Guest On Physical Drive <sup>۳</sup>
- Installing Arch Linux in VMware <sup>۴</sup>

## راه‌اندازی رسانه نصب

ابتدا لازم است که الویت راه‌اندازی در بایوس کامپیوتر شما دستکاری شود تا بتوانید رسانه نصب را به درست راه‌اندازی کنید. برای ورود به بایوس به صورت معمول بایستی یکی از کلیدهای Delete, F1, F2, F11 یا F12 را فشار دهید. این کلید معمولاً در صفحه Post که دقیقاً بعد از روشن شدن کامپیوتر نمایش داده می‌شود مشخص است. در پایه‌ای‌ترین حالت و برای نصب از روی لوح فشرده (CD) بایستی الویت راه‌اندازی بر روی CD or DVD قرار بگیرد. در اکثر مادربردهای موجود در ایران قابلیت انتخاب الویت راه‌اندازی با زدن دکمه F12 در هنگام نمایش اطلاعات سیستم (Post) به شما داده می‌شود.

بعد از راه‌اندازی رسانه نصب گزینه Boot Arch Linux را برای ورود به نصب انتخاب کرده و کلید Enter را بزنید.

**نکته:** شما برای نصب آرچ لینوکس به ۶۴ مگابایت رم در پایه‌ای‌ترین حالت نیاز خواهید داشت.

[https://wiki.archlinux.org/index.php/Arch\\_Linux\\_VirtualBox\\_Guest](https://wiki.archlinux.org/index.php/Arch_Linux_VirtualBox_Guest) ۱

[https://wiki.archlinux.org/index.php/Installing\\_Arch\\_Linux\\_from\\_VirtualBox](https://wiki.archlinux.org/index.php/Installing_Arch_Linux_from_VirtualBox) ۲

[https://wiki.archlinux.org/index.php/VirtualBox\\_Arch\\_Linux\\_Guest\\_On\\_Physical\\_Drive](https://wiki.archlinux.org/index.php/VirtualBox_Arch_Linux_Guest_On_Physical_Drive) ۳

[https://wiki.archlinux.org/index.php/Installing\\_Arch\\_Linux\\_in\\_VMware](https://wiki.archlinux.org/index.php/Installing_Arch_Linux_in_VMware) ۴

## آزمایش این که آیا رایانه شما در حالت UEFI راه‌اندازی شده؟

در صورتی که شما یک مادربرد UEFI داشته باشید و این قابلیت در آن فعال باشد، CD/USB به صورت خودکار هسته آرچ‌لینوکس را اجرا می‌کند (EFISTUB از طریق Gummiboot Boot Manager). برای بررسی این که آیا رایانه شما در این مود راه‌اندازی شده یا خیر ماژول کرنل efivars را بارگزاری کنید. (قبل از chroot) و سپس بررسی کنید که فایل‌هایی در `/sys/firmware/efi/vars/` وجود دارد:

```
# modprobe efivars # before chrooting
# ls -l /sys/firmware/efi/vars/
```

**نکته:** ماژول هسته efivars متغیرهای UEFI را شناسایی و در `/sys/firmware/efi/vars` مستقر می‌کنند. این ماژول در طول پروسه راه‌اندازی به صورت خودکار راه‌اندازی نمی‌شود و تا زمانی که این ماژول بارگزاری و هسته در حالت UEFI بدون پارامتر `noefi` راه‌اندازی شود هیچ فایل‌هایی در `/sys/firmware/efi/vars` ساخته نخواهد شد. این متغیرها در ادامه توسط `efibootmgr` برای اضافه شدن به مدخل بوت‌لودر در منوی راه‌اندازی UEFI تغییر داده می‌شوند. در حالت بایوس، `modprobe` هیچ‌گونه خطایی در مورد ماژول efivars نخواهد داد. راه صحیح برای شناسایی UEFI چک کردن فایل‌های موجود در این مسیر است: `/sys/firmware/efi/vars`

## حل مشکلات در حین راه‌اندازی

- اگر شما از چیپست ویدیویی اینتل استفاده می‌کنند و در زمان راه‌اندازی با یک صفحه سیاه خالی مواجه می‌شوید مشکل به احتمال زیاد مربوط به تنظیمات مد هسته (KMS) می‌شود. یک راه‌حل راه‌اندازی مجدد سیستم (Restart) و فشار دکمه Tab در حین مشاهده منوی بوت است. در اینجا شما در آخر رشته‌ای که ظاهر می‌شود عبارت `nomodeset` را تایپ کنید و Enter را بزنید. راه دوم استفاده از `video=SVIDEO-1:d` است که اگر کار کند تنظیمات مد هسته را غیرفعال نمی‌کند. برای اطلاعات بیشتر صفحه مربوط به اینتل<sup>۲</sup> را مشاهده کنید.
- اگر صفحه سیاه نشد در عوض سیستم در حین عملیات راه‌اندازی زمانی که در حال بارگزاری هسته بودید قفل کرد باز در حین مشاهده منوی بوت Tab را فشار دهید و عبارت

[https://wiki.archlinux.org/index.php/Install\\_from\\_SSH](https://wiki.archlinux.org/index.php/Install_from_SSH) ۵

<https://wiki.archlinux.org/index.php/KMS> ۱

<https://wiki.archlinux.org/index.php/Intel> ۲

acpi=off را در انتهای رشته ظاهر شده تایپ و Enter را بزنید.

## بخش دوم

# نصب

اکنون شما با اعلان خط فرمان مواجه هستید و به صورت خودکار وارد حساب کاربری ریشه شده‌اید.

## تغییر زبان

ترفند: این مرحله به صورت اختیاریست و تنها برای آن دسته از کاربرانی که قصد دارند در فایل‌های پیکربندی به زبان خودشان بنویسند استفاده دارد، اگر شما از نشانه‌های خاص برای پسورد وای-فای خود استفاده کرده‌اید یا می‌خواهید پیغام‌های سیستمی (مثلاً خطاها) را به زبان خود دریافت کنید این امکان می‌تواند برای شما مفید واقع شود.

به صورت پیش‌فرض طرح صفحه‌کلید شما بر روی US تنظیم شده پس اگر شما از کیبورد غیر-US استفاده می‌کنید فرمان زیر را صادر کنید:

```
# loadkeys layout
```

در فرمان بالا *layout* می‌تواند *fr, uk, be-latin1* و غیره باشد. برای مشاهده لیست کامل به صفحهٔ مربوط به آن<sup>۱</sup> مراجعه کنید.

فونت‌ها نیز بایستی تغییر کند چرا که بیشتر زبان‌ها بیش از ۲۶ حرف زبان انگلیسی را شامل می‌شوند در نتیجه اگر شما فونت را تغییر ندهید کارکترها به صورت مربع یا نمادهای دیگر در جلوی شما ظاهر خواهند شد پس دقیقاً دستور زیر را تایپ کنید:

```
# setfont Lat2-Terminus16
```

به صورت پیش‌فرض زبان بر روی انگلیسی-آمریکایی تنظیم شده است. اگر شما قصد عوض کردن زبان را برای پروسه نصب دارید (به عنوان مثال آلمانی) علامت(های) # موجود در جلوی محل مورد نظر در فایل `/etc/locale.gen` را بردارید. لطفاً ورودی UTF-8 را انتخاب کنید. از `Ctrl + X` برای خارج شدن استفاده کنید و زمانی که درخواست ذخیره تغییرات داده شده کلید `Y` و سپس `Enter` را فشار دهید.

```
# nano /etc/locale.gen
```

```
en_US.UTF-8 UTF-8
de_DE.UTF-8 UTF-8
```

```
# locale-gen
# export LANG=de_DE.UTF-8
```

به یاد داشته باشید که `Alt+Shift` (سمت چپ کیبورد) `Keymap` را فعال و غیرفعال می‌کند.

## اتصال به اینترنت

دیمون شبکه `dhcpcd` به صورت خودکار در زمان راه‌اندازی اجرا می‌شود و در صورت وجود یک شبکه با سیم تلاش می‌کند تا با دنیای خارج ارتباط برقرار کند. برای اطمینان از تلاش مؤثر این دیمون از `ping` استفاده کنید:

```
# ping -c 3 www.google.com
```

```
PING www.l.google.com (74.125.132.105) 56(84) bytes of data:
64 bytes from wb-in-f105.1e100.net (74.125.132.105): icmp_req=1 ttl=50 time=17.0 ms
64 bytes from wb-in-f105.1e100.net (74.125.132.105): icmp_req=2 ttl=50 time=18.2 ms
64 bytes from wb-in-f105.1e100.net (74.125.132.105): icmp_req=3 ttl=50 time=16.6 ms
```

```
--- www.l.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
```

```
rtt min/avg/max/mdev = 16.660/17.320/18.254/0.678 ms
```

اگر خروجی برخلاف خروجی بالا عبارت `ping: unknown host` بود متأسفانه تلاش `dhcpcd` بی نتیجه بوده است و شما همانگونه که در ادامه گفته می شود بایستی یک اتصال به صورت دستی آماده کنید.

در غیر این صورت به قسمت آماده سازی درایو ذخیره ساز بروید.

## با سیم

این راه را برای تنظیم یک اتصال با سیم از طریق IP address ایستا دنبال کنید.

اگر کامپیوتر شما به یک شبکه Ethernet متصل شده، در اغلب مواقع شما یک رابط به اسم `eth0` خواهید داشت. اگر شما کارت های شبکه بیشتری دارید (سوی کارت شبکه ای که به صورت درونی در مین برد شما کار شده است) نام های این رابط ها به ترتیب `eth1` و `eth2` و غیره خواهد بود.

شما بایستی این تنظیمات را بدانید:

- Static IP address.
- Subnet mask.
- Gateway's IP address.
- Name servers' (DNS) IP addresses.
- Domain name (مگر این که شما بر روی یک شبکه محلی باشید).

با دستور زیر رابط متصل شده را فعال کنید (به عنوان مثال رابط `eth0`):

```
# ip link set eth0 up
```

آدرس را اضافه کنید:

```
# ip addr add <ip address>/<subnetmask> dev <interface>
```

به عنوان مثال:

```
# ip addr add 192.168.1.2/24 dev eth0
```

برای گزینه‌های بیشتر دستور `man ip` را اجرا کنید.  
گیت‌وی<sup>۱</sup> را به این صورت اضافه کنید:

```
# ip route add default via <ip address>
```

به عنوان مثال:

```
# ip route add default via 192.168.1.1
```

`resolv.conf` را ویرایش کنید و IP آدرس‌های `server name`ها و نام دامنه محلی خود را مشخص کنید:

```
# nano /etc/resolv.conf
```

```
nameserver 61.23.173.5
nameserver 61.95.849.8
search example.com
```

**نکته:** در حال حاضر شما نهایتاً می‌توانید سه `nameserver` را مشخص کنید.

اکنون شما بایستی به خوبی به اینترنت متصل شده باشید. اگر نشده‌اید بهتر است جزئیات بیشتری را بررسی کنید.<sup>۲</sup>

## بی‌سیم

این روش را در صورتی که نیاز به اتصال از طریق شبکه بی‌سیم دارید پی بگیرید.

راه‌اندازها و ابزارهای وایرلس در محیط زنده مدیای نصب در دسترس قرار دارند. یک شناخت خوب از سخت‌افزار وایرلس شما کلید موفقیت شما در پیکربندی و یک اتصال بی‌درد خواهد بود. به خاطر داشته باشید که روشی که در ادامه به آن اشاره می‌کنیم سخت‌افزار وایرلس شما را برای استفاده در محیط زنده مدیای نصب آماده می‌کند. این مراحل بایستی بعد از نصب در سیستم نصب شده هم اجرا تا اتصال شما برقرار شود.

همچنین توجه داشته باشید که این مراحل اختیاری می‌باشد و چنانچه یک اتصال بی‌سیم در این

gateway ۱

[https://wiki.archlinux.org/index.php/Network\\_Configuration](https://wiki.archlinux.org/index.php/Network_Configuration) ۲

مرحله مورد نیاز شما نباشد می‌توانید آن را بعداً در سیستم نصب شده انجام دهید.

**نکته:** در مثال پیش رو، ما از wlan0 به عنوان رابط و linksys برای ESSID استفاده کرده‌ایم. به خاطر داشته باشید که این متغیرها بنا به تنظیمات شما متفاوت می‌باشد.

راه حل پایه‌ای به این صورت خواهد بود:

- (اختیاری) رابط وایرلس خود را شناسایی کنید:

```
# lspci | grep -i net
```

- یا اگر از مدل USB استفاده می‌کنید:

```
# lsusb
```

- مطمئن شوید udev راه‌انداز را بارگیری کرده و راه‌انداز یک رابط هسته وایرلس معمول را با iwconfig ساخته است.

**نکته:** اگر شما خروجی مشابه را دریافت نمی‌کنید در نتیجه راه‌انداز به درستی بارگیری نشده. در این حالت شما بایستی خودتان راه‌انداز را بارگیری کنید. تنظیمات وایرلس را برای اطلاعات بیشتر مشاهده کنید<sup>۱</sup>.

```
# iwconfig
```

```
lo no wireless extensions.
eth0 no wireless extensions.
wlan0 unassociated ESSID:""
      Mode:Managed Channel=0 Access Point: Not-Associated
      Bit Rate:0 kb/s Tx-Power=20 dBm Sensitivity=8/0
      Retry limit:7 RTS thr:off Fragment thr:off
      Power Management:off
      Link Quality:0 Signal level:0 Noise level:0
      Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
      Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

در این نمونه wlan0 در دسترس قرار دارد.

- رابط را بالا بیاورید:



```
# ip link set wlan0 up
```

درصد پایینی از چیپست‌های وایرلس علاوه بر راه‌انداز به سفت‌افزار<sup>۱</sup> نیز احتیاج دارد. اگر چیپست وایرلس شما اینچنین است زمانی که سعی در بالا آوردن رابط دارید با خطایی مشابه این خطا مواجه خواهید شد:

```
# ip link set wlan0 up
```

```
SIOCSIFFLAGS: No such file or directory
```

اگر مطمئن نیستید از `dmesg` برای بررسی درخواست سفت‌افزار توسط چیپست وایرلس خود استفاده کنید. مثالی از خروجی یک چیپست اینتل که نیاز به سفت‌افزار دارد و این درخواست را از هسته در زمان راه‌اندازی انجام داده:

```
# dmesg | grep firmware
```

```
firmware: requesting iwlwifi-5000-1.ucode
```

اگر خروجی مشاهده نمی‌کنید به نظر می‌رسد که چیپست وایرلس شما نیازی به سفت‌افزار ندارد.

**اخطار:** بسته سفت‌افزار چیپست وایرلس (برای کارت‌هایی که نیاز به آن دارند) در `/usr/lib/firmware` (محیط زنده) پیش‌نصب شده است اما بایستی در سیستم اصلی شما برای کارکرد وایرلس بعد از شروع مجدد نصب شود. نصب بسته در ادامه این کتاب آموزش داده شده. مطمئن شوید که هم مازول و هم سفت‌افزار قبل از شروع مجدد نصب شده باشند. تنظیمات وایرلس را اگر شما از نصب سفت‌افزار مورد نیاز چیپست خاص شما اطلاعی ندارید مشاهده کنید.<sup>۲</sup>

• در ادامه از منوی وایرلس `netcfg` برای اتصال استفاده کنید:

```
# wifi-menu wlan0
```

در این مرحله شما بایستی به شبکه متصل شده باشید. اگر نشده‌اید تنظیمات وایرلس را ببینید.<sup>۳</sup>

firmware ۱

[https://wiki.archlinux.org/index.php/Wireless\\_Setup](https://wiki.archlinux.org/index.php/Wireless_Setup) ۲

[https://wiki.archlinux.org/index.php/Wireless\\_Setup](https://wiki.archlinux.org/index.php/Wireless_Setup) ۳

## ISDN یا xDSL (PPPoE), analog modem

اگر روتر<sup>۱</sup> شما در حالت پل<sup>۲</sup> قرار دارد دستور زیر را اجرا کنید:

```
# pppoe-setup
```

- نام کاربری که سرویس دهنده به شما اختصاص داده است را وارد کنید.
  - برای eth0 دکمه برگشت را فشار دهید.
  - کلید برگشت را برای no فشار دهید (تازمانی که نیاز هست).
  - کلمه server بنویسید.
  - 1 را برای دیوارآتش<sup>۳</sup> فشار دهید.
  - کلمه عبور خود، که توسط سرویس دهنده مشخص شده را وارد کنید.
  - دکمه ۷ را برای خاتمه فشار دهید.
- برای استفاده از این تنظیمات و اتصال به سرویس دهنده دستور زیر را صادر کنید:

```
# pppoe-start
```

شاید بخواهید که resolv.conf را ویرایش کنید:

```
# echo nameserver 8.8.8.8 > /etc/resolv.conf
```

## در پی پروکسی

اگر شما در پشت یک سرور پروکسی هستید نیاز به صدور یک متغیر محیطی http\_proxy و/یا ftp\_proxy به شکل زیر دارید.

```
export http_proxy=http://<http_proxy_address>:<proxy_port>
export ftp_proxy=ftp://<ftp_proxy_address>:<proxy_port>
```

به مثال‌های زیر دقت کنید:

```
export http_proxy=http://10.203.0.1:5187/
```

Router	۱
Bridge	۲
Firewall	۳

```
export https_proxy=http://10.203.0.1:5187/
export ftp_proxy=http://10.203.0.1:5187/
export no_proxy="localhost,127.0.0.1,localaddress,.localdomain.com"
```

به عنوان مثال برای تنظیم پروکسی برنامه `your-freedom` دستور زیر را صادر کنید:

```
export http_proxy="http://localhost:8080"
```

یا

```
export http_proxy="http://127.0.0.1:8080"
```

برای خروج از پس پروکسی دستور زیر را صادر کنید:

```
unset http_proxy
```

## آماده کردن درایو ذخیره ساز

اخطار: پارتیشن بندی اطلاعات شما را از بین خواهد برد. کاملاً توجه داشته باشید که قبل از این کار از اطلاعات حیاتی خود نسخه پشتیبان تهیه کنید.

## پارتیشن بندی هارد دیسک (اطلاعات عمومی)

### انواع پارتیشن ها

پارتیشن بندی هارد دیسک، قسمت های به خصوصی (پارتیشن ها) را در داخل دیسک شما ایجاد می کند که بسته به فایل سیستم شان به صورت یک دیسک به خصوص دیده می شوند و رفتاری مشابه دارند.

سه نوع پارتیشن وجود دارد:

- Primary
- Extended
- Logical

پارتیشن Primary می تواند قابلیت راه اندازی داشته باشد، به عبارت دیگر می تواند bootable باشد.

این دسته از پارتیشن‌ها به ۴ عدد بر روی هر دیسک یا حجم raid محدود شده‌اند. اگر نیاز به بیش از ۴ پارتیشن بر روی یک دیسک دارید نوع Extended که در برگیرنده پارتیشن‌های logical می‌شود می‌تواند به شما کمک کند.

پارتیشن Extended به خودی خود قابل استفاده نیست و فقط در برگیرنده پارتیشن‌های logical می‌شود. یک هارددیسک تنها یک پارتیشن از نوع گسترده یا Extended می‌تواند داشته باشد. توجه داشته باشید این پارتیشن به عنوان یک پارتیشن Primary شناخته می‌شود. به عبارت دیگر اگر شما قصد ساخت بیش از ۴ پارتیشن را داشته باشید می‌توانید حداکثر ۳ پارتیشن از نوع Primary و یک پارتیشن از نوع Extended حاوی هر تعداد پارتیشن logical که می‌خواهید، داشته باشید.

زمانی که در حال پارتیشن‌بندی هارد دیسک خود هستید، پارتیشن‌های Primary از sda1 تا sda3 شناخته می‌شوند. پارتیشن Extended یا گسترده هم sda4 را برای خود انتخاب می‌کند و بقیه پارتیشن‌های ساخته شده در پارتیشن گسترده که از نوع logical هستند، sda5 و sda6 و ... را به خود اختصاص می‌دهند.

## طرح پارتیشن

طرح پارتیشن یک دیسک دارای ترجیحات زیادیست. انتخاب هر کاربر بسته به احتیاجات و عاداتش می‌تواند متفاوت باشد.

کاندیدهای فایل سیستم برای داشتن پارتیشن‌های جدا به ترتیب (- اهمیت) به شرح ذیل می‌باشند:

- / (ریشه): فایل سیستم ریشه نسبت به بقیه فایل سیستم‌ها از الویت بالاتری برخوردار است و لازمه یک توزیع گنو/لینوکس وجود این فایل سیستم است. تمام فایل‌ها و مسیرها حتی اگر در دیوایس‌های فیزیکی دیگر هم باشند در زیر مسیر ریشه (/) نمایان می‌شوند. با این اوصاف وجود هیچ یک از مسیرهای زیرین ریشه به خودی خود نسبت به خود ریشه، دارای ارجحیت برای داشتن پارتیشن جدا نخواهند بود.
- /boot: این مسیر شامل ایمیج‌های هسته، ramdisk، فایل پیکربندی راه‌انداز یا بوت‌لودر و bootloader stages می‌شود. /boot همچنین اطلاعاتی که قبل از اجرای هسته باید اجرا شود را در خود نگه می‌دارد. همچنین شامل master boot sectors و sector map files نیز می‌شود. /boot کاملاً برای راه‌اندازی سیستم لازم است اما می‌تواند پارتیشن جدایی برای خود نداشته باشد و به صورت زیر مجموعه‌ای از ریشه وجود داشته باشد.
- /home: در بردارنده زیر دایرکتوری‌های مربوط به هر کاربر سیستم، برای نگه‌داری اطلاعات

گوناگون شخصی مانند فایل‌های پیکربندی مربوط به کاربران برای برنامه‌ها و ابزارهای مختلف است.

- **/tmp**: وجود این مسیر برای برنامه‌هایی که فایل‌های موقت مانند `lck` دارند الزامیست. این کار از ایجاد نمونه‌های متعدد از فایل مربوط به برنامه‌ها تا زمانی که کار برنامه تکمیل شود جلوگیری می‌کند و در زمان تکمیل کار برنامه، آن را حذف می‌نماید. برنامه‌ها هم نباید انتظار داشته باشند فایل‌هایی که در این مسیر قرار می‌گیرند برای استفاده‌های بعد نگه‌داری شوند چرا که به صورت معمول بعد از هر راه‌اندازی مجدد سیستم، کلیه فایل‌های موجود در این دایرکتوری و همچنین زیر دایرکتوری‌های آن حذف خواهند شد.
- **/var**: این مسیر شامل اطلاعات متغیری است؛ جایی برای دایرکتوری و فایل‌های مدیریتی، اطلاعات مربوط به گزارشات، کش پک‌من، درخت `ABS` و ... می‌باشد. `/var` به وجود آمده برای این که `/usr` بتواند به صورت فقط خواندنی سوار شود. هر چیزی که در حال انجام عملیات سیستمی (بر خلاف نصب و نگه‌داری نرم‌افزاری) به `/usr` وارد می‌شود بایستی در مسیر `/var` ساکن باشد.

**اخطار:** در کنار `/boot` دایرکتوری‌های ضروری برای راه‌اندازی سیستم عبارتند از: `'/lib', '/etc', '/bin', '/sbin'` که نباید در پارتیشنی غیر از پارتیشن ریشه ساکن شوند. به عبارت دیگر مسیرهای نام برده بایستی نقطه اتصال جداگانه داشته باشند.

چندین مزیت برای استفاده از فایل سیستم‌های جداگانه نسبت به مخلوط کردن آن‌ها با هم وجود دارد که به قرار زیر است:

- **امنیت در دسترسی:** هر فایل سیستم می‌تواند در `/etc/fstab` به صورت‌های `nodev`, `nosuid` و `noexec, readonly` و غیره پیکربندی شود.
- **پایداری:** یک کاربر و یا برنامه مشکل‌دار در صورت داشتن مجوزهای لازم می‌تواند کل فایل سیستم را با مشکل مواجه کند. برنامه‌ها و فایل‌های حیاتی که در فایل سیستم‌های جداگانه نگه‌داری می‌شوند می‌تواند از این خطر مصون باشند.
- **سرعت:** یک فایل سیستم که درگیر تمام کارهای سیستم است و به صورت متناوب توسط قسمت‌های مختلف بر روی آن نوشته می‌شود، می‌تواند محل مناسبی برای فایل‌های تکه‌تکه و از هم جدا باشد که در نهایت باعث کند شدن سرعت عمل‌کرد سیستم می‌شود. یکی از راه‌های جلوگیری از بوجود آمدن فایل‌های تکه‌تکه وجود فضای لازم در فایل سیستم به

صورتی که مطمئن شویم پارتیشن مورد نظر به مرحله خطر برای کمبود فضا نمی‌رسد، است. فایل سیستم‌های جدا باز هم از این خطر در امان هستند و هر کدام به خوبی می‌توانند یک پارچه‌سازی شوند.

- امنیت فایل سیستم: اگر یک فایل سیستم خراب شود بقیه فایل سیستم‌های جدا در امنیت کامل به سر خواهند برد.
  - تطبیق پذیری: اشتراک فایل‌های بین سیستم‌های مختلف زمانی که از فایل سیستم‌های مجزا استفاده شود مناسب‌تر خواهد بود. نوع هر فایل سیستم می‌تواند با توجه به فایل‌هایی که در آن نگهداری می‌شود و مورد استفاده از آن متفاوت باشد.
- در این مثال ما قصد ساخت پارتیشن‌های مجزا برای ریشه، `/var`، خانه و `swap` را داریم.

**تذکر:** `/var` حاوی فایل‌های کوچک بی‌شماری می‌شود. این موضوع بایستی در زمان انتخاب نوع فایل سیستم برای آن لحاظ شود (اگر پارتیشن جدایی برای آن در نظر گرفته شود).

## هر پارتیشن چقدر بزرگ باشد؟

بهترین پاسخ به این سؤال به احتیاجات فردی هر کس مرتبط است. شاید شما بخواهید به صورت بسیار ساده یک پارتیشن برای ریشه به علاوه یک پارتیشن برای `swap` داشته باشید و یا حتی فقط یک پارتیشن برای ریشه بدون `swap` داشته باشید و یا به مانند مثال ما چندین پارتیشن مجزا برای نقطه‌های اتصال جدا داشته باشید.

- فایل سیستم ریشه در مثال ما حاوی دایرکتوری `/usr` که بسته به نرم‌افزارهایی که بر روی سیستم نصب شده حجیم می‌شود، است. با این تفاسیر فضای ۱۵ تا ۲۰ گیگابایت برای بیشتر کاربران به عنوان فضای ریشه کفایت می‌کند.
- فایل سیستم `/var` شامل دیگر اطلاعات، درخت `ABS`، و کش پک‌من می‌شود. نگهداری بسته‌های گرفته شده می‌تواند مفید باشد؛ و برای ما امکان دانگرید راحت بسته‌ها را در صورت لزوم فراهم می‌کند. فضای اشغال شده در `/var` به مرور افزایش پیدا می‌کند، بسته‌های داخل کش پک‌من باگذشت زمان زیاد و زیادتر می‌شوند و می‌توانیم آن‌ها را زمانی که به بسته‌های اضافی نیازی نداریم پاک‌کنیم. اگر شما از یک `SSD` استفاده می‌کنید شاید بهتر باشد که `/var` را بر روی `HDD` خود قرار دهید و ریشه (`/`) و خانه (`/home`) را بر روی `SSD` سوار کنید تا از خواندن/نوشتن‌های بی‌مورد بر روی `SSD` پیشگیری کرده باشید. برای سیستم‌های دسکتاپ اندازه ۸ تا ۱۲ گیگابایت بایستی کافی باشد. همچنین بسته به

نرم افزارهایی که می خواهید نصب کنید می تواند بزرگ تر باشد. برای سرور بایستی این فایل سیستم نسبتاً بزرگ تر انتخاب شود.

- فایل سیستم خانه جایی است که اطلاعات، فایل های دانلود شده و چند رسانه ای های کاربر در آن قرار می گیرد. برای یک سیستم دسکتاپ این فایل سیستم به طور معمول بزرگ ترین فایل سیستم است. به یاد داشته باشید که چنانچه بخواهید مجدداً آرچ نصب کنید تمامی اطلاعات داخل این پارتیشن در صورت فرمت نشدن دست نخورده باقی خواهد ماند، البته این امر مستلزم استفاده از خانه بر روی یک پارتیشن مجزاست.

- ۲۵ درصد فضای اضافه تر به فایل سیستم ها برای مواجه نشدن با کمبود فضا و رخدادهای پیش بینی نشده و همچنین جلوگیری از تکه تکه شدن فایل ها اختصاص دهید.

با توجه به راهنمای فوق بهتر است که پارتیشن ریشه ۱۵ گیگابایت، پارتیشن var ده گیگابایت، swap یک گیگابایت و پارتیشن خانه بقیه فضای دیسک را به خود اختصاص دهد.

## فایل سیستم ها (اطلاعات عمومی)

نوع یک فایل سیستم به ترجیحات و استفاده های شخصی هر فرد بستگی دارد و متغیر است. هر کدام مزیت ها، معایب و ویژگی های مختص به خود را دارند. در اینجا یک نگاه مختصر بر فایل سیستم های پشتیبانی شده می اندازیم:

- **ext2 Second Extended Filesystem**: فایل سیستم قدیمی گنو/لینوکس. بسیار پایدار است اما از journaling و barriers پشتیبانی نمی کند. مشکل عدم پشتیبانی از journaling زمانی نمود پیدا می کند که سیستم شما دچار سانحه (کرش) شود یا به صورت غیر معمول خاموش شود (رفتن برق) و ممکن است با از دست دادن اطلاعات شما همراه باشد. این فایل سیستم برای ریشه و خانه مناسب نیست ولی به راحتی قابل تبدیل به ext3 است.

- **ext 3 Third Extended Filesystem**: همان فایل سیستم ext2 با قابلیت پشتیبانی از journaling و write barriers. همانند ext2 بسیار پایدار و کامل است.

- **ext4 Fourth Extended Filesystem**: سازگار با ext2 و ext3 با قابلیت پشتیبانی از یک اگزابایت برای درایوها و ۱۶ ترابایت برای هر فایل. افزایش محدودیت ۳۲۰۰۰ زیر مسیر (subdirectory) در ext3 به ۶۴۰۰۰۰ زیر مسیر. قابلیت یک پارچه سازی آنلاین.

- **ReiserFS (V3) Hans Reiser's high-performance journaling FS**: از متد

بسیار جذابی برای کار بر روی داده‌ها بر مبنای الگوریتم خلاق و نامنظم استفاده می‌کند. ReiserFS بسیار سریع است مخصوصاً زمانی که با تعداد زیادی از فایل‌های کوچک سرکار داشته باشید. ReiserFS در قالب‌بندی (formatting) هم بسیار سریع است اما نسبتاً برای اتصال (mounting) کند است. کاملاً بالغ و پایدار می‌باشد. ReiserFS (V3) در حال حاضر توسعه داده نمی‌شود. به طور کلی گزینه مناسبی برای /var می‌باشد.

- **JFS IBM's Journaled FileSystem**: اولین فایل سیستم که از journaling پشتیبانی می‌کرد. JFS قبل از این که به گنو/لینوکس بیاید سال‌ها در IBM AIX® OS مورد استفاده قرار می‌گرفت. JFS در حال حاضر از حداقل منابع CPU در بین فایل سیستم‌های گنو/لینوکس استفاده می‌کند. همچنین در زمینه قالب‌بندی (formatting)، اتصال (mounting) و fsck (چک کردن فایل سیستم) بسیار سریع است. در کل این فایل سیستم عمل‌کرد بسیار خوبی دارد به خصوص در اتصال با زمان‌بند ورودی و خروجی. JFS مانند ext یا ReiserFS پشتیبانی نمی‌شود ولی بسیار بالغ و پایدار است.
- **XFS**: یک فایل سیستم دیگر با قابلیت پشتیبانی از journaling که توسط Silicon Graphics برای IRIX OS توسعه داده شده و به گنو/لینوکس برده شده است. XFS برای فایل‌ها و فایل سیستم‌های بزرگ بسیار سریع است. همچنین در قالب‌بندی و اتصال سرعت بالایی دارد. در آزمایش‌های انجام شده نسبت به فایل سیستم‌های دیگر در مورد فایل‌های کوچک کندتر عمل کرده است. این فایل سیستم بالغ قابلیت یک‌پارچه‌سازی آنلاین را دارد.
- **vfat**: فایل سیستم Fat با پشتیبانی از نام‌های بزرگ برای فایل‌ها.
- **Btrfs**: این فایل سیستم با نام "Better FS" نیز شناخته می‌شود که با اصول جدید و ویژگی‌های قدرتمند شبیه ZFS ارائه شده است که شامل snapshotها، multi-disk striping و mirroring، همچنین پشتیبان‌گیری اضافی، checksumها، فشرده‌سازی روی هوا (که می‌تواند عمل‌کرد و صرفه‌جویی در فضا را بهبود ببخشد) و ویژگی‌های دیگر می‌شود. در ژانویه ۲۰۱۱ هنوز این فایل سیستم به عنوان یک فایل سیستم ناپایدار معرفی شده اما به صورت آزمایشی وارد خطوط اصلی هسته شده است. به نظر می‌رسد Btrfs فایل سیستم بعدی لینوکس‌ها خواهد بود. اکنون این فایل سیستم به عنوان فایل سیستم پارتیشن ریشه توسط نصاب توزیع‌های بزرگ گنو/لینوکس پیشنهاد می‌شود.
- **Nilfs2**: یک فایل سیستم دیگر برای لینوکس که توسط NTT توسعه داده شده و تحت مجوز GPL می‌باشد. NILFS یک فایل سیستم log-structured می‌باشد.



## فضای Swap

لینوکس، حافظه اصلی را به قسمت‌هایی به اسم پیج‌ها تقسیم می‌کند. Swapping به عملیاتی گفته می‌شود که یک پیج از حافظه اصلی بر روی بخشی از پیش تعیین شده بر روی هارد به اسم فضای Swap کپی شود تا حافظه اصلی برای انجام امور دیگر خالی شود. ترکیب اندازه حافظه اصلی و فضای Swap مقدار حافظه موجود شما را تشکیل می‌دهد.

فضای Swap می‌تواند یک پارتیشن جداگانه یا یک فایل باشد. برای کاربرانی که حافظه اصلی آن‌ها کمتر از یک گیگابایت است داشتن فضای Swap توصیه می‌شود.

### فضای swap من چقدر باید بزرگ باشد؟

سؤال کلیدی این است که چقدر؟ نسخه‌های قدیمی سیستم‌عامل‌های گنو/لینوکس به فضای swap با بزرگی ۲ تا ۳ برابر حافظه اصلی احتیاج داشتند اما امروزه با عوض شدن تکنیک‌های مبادله و ارزان شدن رم نیازی به این مقدار swap احساس نمی‌شود. به جدول زیر برای یک حالت نسبتاً عمومی دقت کنید:

Desktop	
مقدار فضای RAM	مقدار فضای SWAP
۵۱۲ مگابایت	۱ گیگابایت
۱ گیگابایت	۳ گیگابایت
۲ گیگابایت	۲ گیگابایت
۴ گیگابایت	۴ گیگابایت (فضای Swap غیرلازم است)
۸ گیگابایت	۴ گیگابایت (فضای Swap غیرلازم است)
۱۶ گیگابایت	۸ گیگابایت (فضای Swap غیرلازم است)

برای سرور هرچقدر فضای swap کمتر باشد (و از SSD استفاده شود) بهتر است. معمولاً فضایی معادل نصف اندازه حافظه اصلی را برای swap در نظر می‌گیرند ولی توجه داشته باشید همیشه فضای swap را بررسی کنید تا در صورت لزوم آن را افزایش دهید.

## اضافه کردن Swap File

تذکر: توجه داشته باشید برای ساخت و استفاده از این فایل با دستورات زیر بایستی در سیستم نصب شده آرچ وارد شده باشید. در نتیجه بهتر است ادامه آموزش را پی بگیرید و بعد از نصب آرچ و وارد شدن به سیستم اصلی اقدام به ساخت آن نمائید.

بهترین راه برای داشتن فضای Swap استفاده از فایل Swap می‌باشد. به این منظور با دسترسی ریشه فرمان `fallocate` را برای ساخت فایل Swap با اندازه‌ای که تعیین کرده‌اید (M = Megabytes, G = Gigabytes) اجرا کنید (در مثال ما ۵۱۲ مگابایت در نظر گرفته شده):

```
# fallocate -l 512M /swapfile
```

یا

```
# dd if=/dev/zero of=/swapfile bs=1M count=512
```

اکنون نوبت به تنظیم دسترسی‌هاست:

```
# chmod 600 /swapfile
```

در این مرحله بایستی فایل را قالب‌بندی کنید:

```
# mkswap /swapfile
```

در ادامه آن را فعال کنید:

```
# swapon /swapfile
```

و در نهایت آن را به فایل `/etc/fstab` اضافه کنید:

```
# nano /etc/fstab
```

```
.
.
.
/swapfile    none    swap    defaults    0    0
```

## حذف Swap File

با دسترسی ریشه دستور زیر را صادر کنید:

```
# swapoff -a
```

و در نهایت فایل را حذف کنید:

```
# rm -rf /swapfile
```

## پارتیشن بندی

به کاربران تازه کار توصیه می شود که از برنامه های با رابط گرافیکی به منظور انجام این مرحله استفاده کنند. Gparted یک برنامه خوب است که دیسک زنده<sup>۱</sup> برای انجام این کار در اختیار شما قرار می دهد و همچنین در اکثر توزیع های گنو/لینوکس مانند اوبونتو در دسترس است. بایستی در ابتدا یک درایو را پارتیشن بندی و سپس فرمت کنید. در ادامه با شروع مجدد سیستم به دیسک زنده آرچ بازگردید.

در هر مرحله از نصب شما می توانید یک فایل Swap ایجاد و تنظیم کنید در نتیجه نیازی به تصمیم گیری در مورد این که Swap چه حجمی داشته باشد در این مرحله ضروری نیست. همچنین شما می توانید به جای فایل از یک پارتیشن مخصوص Swap استفاده کنید ولی توصیه ما به شما فایل Swap است.

اگر کار شما با پارتیشن بندی تمام شده به قسمت سوار کردن پارتیشن ها بروید. اما در صورتی که قصد پارتیشن بندی از طریق دیسک زنده آرچ را دارید به مثال زیر توجه کنید.

- برنامه های زیر در آرچ لینوکس برای پارتیشن بندی در دسترس شما قرار دارند:
- `gdisk` و `cgdisk` (تنها جدول پارتیشن ها از نوع GPT را پشتیبانی می کند).
- `fdisk` و `cdisk` (تنها جدول پارتیشن ها از نوع MBR را پشتیبانی می کند).
- `Parted` هر دو را پشتیبانی می کند.

در مثال ما از `cdisk` استفاده شده ولی به راحتی برای `cgdisk` هم قابل استفاده است.

### نکته مربوط به راه‌اندازی UEFI:

- اگر شما مادربرد UEFI دارید نیاز به ساخت پارتیشن سیستم UEFI اضافی خواهید داشت.<sup>۱</sup>
- پیشنهاد می‌شود برای راه‌اندازی UEFI همیشه از GPT استفاده کنید چرا که سفت‌افزارهای UEFI در پاره‌ای از مواقع راه‌اندازی UEFI-MBR را مسدود می‌کند.

### نکته مربوط به پارتیشن‌بندی GPT:

- اگر سیستم شما با ویندوز به صورت بوت‌دوگانه نیست به شما پیشنهاد می‌شود که از GPT به جای MBR با توجه به مزایای آن<sup>۲</sup> استفاده کنید.
- اگر شما یک مادربرد BIOS دارید (یا قصد راه‌اندازی در مد بایوس دارید) و می‌خواهید که گراب را بر روی یک پارتیشن GPT تنظیم کنید شما به یک پارتیشن راه‌اندازی بایوس<sup>۳</sup> ۲ مگابایتی نیاز دارید. Syslinux نیاز به این کار ندارد.
- برخی سیستم‌های بایوس ممکن است با GPT مشکل داشته باشند<sup>۴</sup> و<sup>۵</sup>.

**نکته:** اگر شما قصد نصب بر روی یک حافظه فلش را دارید به قسمت نصب آرچ‌لینوکس بر روی حافظه فلش مراجعه کنید در غیر این صورت به سراغ اجرای cfdisk بروید.

## نصب آرچ‌لینوکس بر روی حافظه فلش

- شما به یک حافظه فلش با حداقل ۲ گیگابایت فضا نیاز دارید. در ادامه راه‌های مختلفی برای نصب آرچ بر روی یک حافظه فلش را بیان خواهیم کرد.
- اگر شما در آرچ‌لینوکس شده هستید اسکریپت آرچ را نصب و با توجه به راهنما اجرا کنید، درست به مانند روشی که برای یک ISO انجام می‌دهید ولی در اینجا نبایستی از `/dev/sda` استفاده کنید. با به کارگیری `lsblk` نام `/dev/sd*` را بدست آورده و آماده نصب شوید.

<sup>۱</sup> [https://wiki.archlinux.org/index.php/Unified\\_Extensible\\_Firmware\\_Interface#Create\\_an\\_UEFI\\_System\\_Partition\\_in\\_Linux](https://wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface#Create_an_UEFI_System_Partition_in_Linux)

<sup>۲</sup> <https://wiki.archlinux.org/index.php/GPT>

<sup>۳</sup> [https://wiki.archlinux.org/index.php/GRUB#GPT\\_specific\\_instructions](https://wiki.archlinux.org/index.php/GRUB#GPT_specific_instructions)

<sup>۴</sup> <http://mjpg59.dreamwidth.org/8035.html>

<sup>۵</sup> <http://rodsbooks.com/gdisk/bios.html>

**اخطار:** اگر شما به اشتباه `/dev/sda` را فرمت کنید با اطلاعات موجود بر روی هارد دیسک (و نه حافظه فلش) بهتر است خداحافظی کرده باشید چرا که آن‌ها پاک خواهد شد.

- به مانند نصب آرچ بر روی یک دیسک سخت شما می‌توانید با استفاده از همان مدیای نصب و طبق راهنمای اصلی، آرچ‌لینوکس را بر روی یک حافظه فلش نصب کنید.
- اگر شما یک لینوکس دیگر (نه الزاماً آرچ‌لینوکس) دارید، می‌توانید از راهنمای نصب از طریق لینوکس نصب شده این کار را انجام دهید<sup>۱</sup>.

### نصب

نصب بر روی یک حافظه فلش مسیر عادی نصب را طی می‌کند و تنها در موارد زیر متفاوت است:

- اگر در `fdisk` با خطای "Partition ends in the final partial cylinder" مواجه شدید تنها راه ناپود کردن تمام پارتیشن‌های موجود بر روی درایو است. یک ترمینال جدید باز کنید (`Alt + F2`)، دستور `fdisk /dev/sdX` را صادر کنید (`sdX` حافظه فلش شماست). جدول پارتیشن را پرینت کنید (`p`)، بررسی کنید که مشکلی وجود ندارد و آن را پاک کنید (`d`) و در نهایت اطلاعات را بنویسد (`w`). اکنون به `fdisk` بازگردید.
- اطلاعات خود را در زمینه کمینه کردن نوشتن/خواندن بر روی SSDها<sup>۲</sup> افزایش دهید و برای نصب به کار ببندید. به عنوان مثال استفاده از گزینه `noatime` در `fstab`. به طور خلاصه استفاده از فایل سیستم `ext4` بدون حالت `journal` به نظر خوب می‌آید. به یاد داشته باشید که اطلاعات در یک فلش تعداد دفعات محدودی قابل نوشتن است و فایل سیستم‌های `journaling` باعث افزایش نوشتن و کم شدن عمر حافظه فلش خواهند شد. در ضمن بهتر است فکر `swap` را نیز از ذهن خود دور کنید. توجه داشته باشید که این نکات برای حافظه فلش مورد استفاده است و برای یک هارد دیسک متصل به `USB` صدق نمی‌کند.
- قبل از ساخت رم‌دیسک `linux -p mkinitcpio #` در فایل `/etc/mkinitcpio.conf`، هوک<sup>۳</sup> `block` را به آرایه‌های هوک درست بعد از `udev` اضافه کنید. این کار برای بارگذاری ماژول‌های مورد نیاز ضروری است.

<sup>۱</sup> [https://wiki.archlinux.org/index.php/Install\\_from\\_Existing\\_Linux](https://wiki.archlinux.org/index.php/Install_from_Existing_Linux)

<sup>۲</sup> [https://wiki.archlinux.org/index.php/SSD#Tips\\_for\\_Minimizing\\_SSD\\_Read.2FWrites](https://wiki.archlinux.org/index.php/SSD#Tips_for_Minimizing_SSD_Read.2FWrites)

<sup>۳</sup> hook

```
HOOKS="... udev block..."
```

## تنظیمات

- مطمئن شوید که اطلاعات موجود در `fstab` در مورد پارتیشن ریشه (`/`) و بقیه پارتیشن‌هایی که ممکن است بر روی حافظه فلش موجود باشد، درست است. اگر قصد استفاده از این آرچ‌لینوکس را در دستگاه‌های مختلف دارید بهتر است که اطلاعات موجود در `fstab` بر اساس UUID پارتیشن‌های مختلف باشد تا با اختلال در شناسایی پارتیشن مواجه نشوید.
- در فایل پیکربندی گراب نکات زیر را مد نظر قرار دهید:

۱. زمانی که گراب بر روی حافظه فلش نصب می‌شود، حافظه `hd0,0` خواهد بود.

۲. در نسخه کنونی گراب ۲ از `uuid` برای شناسایی استفاده می‌شود.

با استفاده از شناسه پارتیشن:

```
root (hd0,0)
kernel /boot/vmlinuz-linux root=/dev/sda1 ro
initrd /boot/initramfs-linux.img
```

با استفاده از برچسب (Label):

```
root (hd0,0)
kernel /boot/vmlinuz-linux root=/dev/disk/by-label/Arch ro
initrd /boot/initramfs-linux.img
```

با استفاده از UUID:

```
root (hd0,0)
kernel /boot/vmlinuz-linux root=/dev/disk/by-uuid/3a9f8929-627b-4667-9db4-388c4eaa9fa ro
initrd /boot/initramfs-linux.img
```

## پارتیشن‌بندی با `fdisk`

برای اجرای `fdisk` بر روی هارد دیسک ابتدایی (`sda`) دستور زیر را صادر کنید:

```
# fdisk /dev/sda
```

ما برای این قسمت دو مثال متفاوت را ذکر خواهیم کرد. در مثال اول با یک هارد دیسک خام

مواجه هستیم که تمامی فضای آن را به آرچ لینوکس اختصاص خواهیم داد و در مثال دوم آرچ لینوکس را بر روی سیستمی که دارای سه پارتیشن NTFS شامل سیستم عامل ویندوز در پارتیشن اول خود می شود نصب خواهیم کرد و یکی از پارتیشن های NTFS را پس از حذف به آرچ لینوکس اختصاص خواهیم داد.

### مثال اول: نصب آرچ بر روی دیسک خام

با انتخاب دیسک مقصد (دیسکی که قرار است آرچ لینوکس بر روی آن نصب شود) در دستور فوق ما دیسک اول یا همان sda را انتخاب کرده ایم] برای پارتیشن بندی دستی باز می شود.

```

cfdisk (util-linux 2.19.1)

Disk Drive: /dev/sda
Size: 125123428352 bytes, 125.1 GB
Heads: 255 Sectors per Track: 63 Cylinders: 15212

-----
Name          Flags      Part Type  FS Type      [Label]      Size (MB)
-----
Pri/Log      Free Space                125123.43*

-----

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]

Create new partition from free space_
    
```

تصویر ۱

با کلیدهای چپ و راست بین گزینه های پایین (Help, New, Print, ...) و با کلیدهای بالا و پایین بین پارتیشن های مختلف جابجا شوید.

فضای دیسک کاملاً خالیست در نتیجه برای ساخت اولین پارتیشن New را انتخاب و Enter را بزنید. در این مرحله با ۳ گزینه روبرو می شوید که درباره آن ها در بخش انواع پارتیشن ها گفته شد. در این مثال ما قصد ساخت چهار پارتیشن از نوع Primary و اختصاص تمام فضای دیسک به آرچ لینوکس را داریم. بعد از انتخاب گزینه Primary برنامه در مورد اندازه این پارتیشن از ما سؤال خواهد کرد. ما این پارتیشن را برای ریشه در نظر گرفته ایم پس پانزده هزار مگابایت را برای آن انتخاب کردیم. توجه داشته باشید که اندازه وارد کرده را به مگابایت بنویسید. در قدم بعدی از ما در

مورد مکان این پارتیشن سؤال خواهد شد که با انتخاب Beginning آن را از اول دیسک سخت خود ایجاد می‌کنیم. با کلیدهای چپ و راست گزینه Type را انتخاب و نوع فایل سیستم ساخته شده را به فایل سیستم لینوکس (عدد ۸۳) تغییر می‌دهیم. با توجه به این که پارتیشن ساخته شده در بردارنده مسیر /boot/ نیز می‌شود پرچم Bootable را بر روی آن با انتخاب گزینه Bootable و فشار کلید Enter می‌زنیم.

اکنون نوبت به ساخت پارتیشن برای /var/ رسیده است. ابتدا با زدن کلید پایین از روی پارتیشن مربوط به ریشه، به فضای خالی دیسک منتقل می‌شویم سپس مراحل ساخت را مانند پارتیشن مربوط به ریشه دنبال می‌کنیم و نوع آن را نیز ۸۳ انتخاب می‌کنیم اما پرچم Bootable را برای این پارتیشن انتخاب نمی‌کنیم. اندازه این پارتیشن در مثال ما ده هزار مگابایت است.

در قدم بعدی پارتیشن مربوط به swap را با اندازه هزار مگابایت می‌سازیم. نوع این پارتیشن با نوع دو پارتیشن قبلی فرق دارد و بایستی ۸۲ (Linux swap / Solaris) را برای آن انتخاب کنیم. در انتها نوبت ساخت پارتیشن خانه با اندازه باقی‌مانده دیسک سخت است. نوع این پارتیشن هم مانند ریشه و /var/ از نوع ۸۳ می‌باشد.

اکنون ۴ پارتیشن Primary ساخته‌ایم و طبیعتاً جایی برای پارتیشن پنجم بر روی این دیسک وجود ندارد؛ در مثال ما، نیازی به وجود پارتیشن پنجم نخواهد بود.

sda1 را برای ریشه، sda2 را برای /var/، همچنین sda3 را برای swap و sda4 را برای خانه ایجاد کرده‌ایم (تصویر ۲).

```

cfdisk (util-linux 2.19.1)

Disk Drive: /dev/sda
Size: 125123428352 bytes, 125.1 GB
Heads: 255 Sectors per Track: 63 Cylinders: 15212

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot       Primary   Linux         15002.92
sda2      Primary   Primary   Linux         10001.95
sda3      Primary   Primary   Linux swap / Solaris 1003.49
sda4      Primary   Primary   Linux         99115.10*

[ Bootable ] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition_
    
```



نام پارتیشن‌های ایجاد شده و مقصد آن‌ها را برای اتصال در مرحله بعدی در گوشه‌ای یادداشت می‌کنیم.

با زدن کلیدهای چپ و راست گزینه Write را انتخاب می‌کنیم و برای تأیید yes را تایپ می‌کنیم و Enter را می‌زنیم. متوجه باشید که انجام این عمل اطلاعات موجود بر روی دیسک شما را با اطلاعات جدید بازنویسی می‌کند. گزینه Quit را برای خروج از برنامه انتخاب کنید. در ادامه Done را برای تأیید بزنید.

تذکر: بعد از آخرین تغییرات هسته لینوکس که شامل ماژول‌های libata و PATA می‌شود تمامی درایوهای SATA، IDE و SCSI به عنوان sdx شناخته می‌شوند و نمایش اینگونه آن‌ها طبیعی است و جای نگرانی نیست.

نکته: در صورتی که ترجیح می‌دهید، دوباره مراحل را انجام دهید و از تنظیماتی که اعمال کرده‌اید ناراضی هستید کفایت با Quit از برنامه بدون این که Write را انجام دهید خارج شوید و cfdisk را مجدداً اجرا کنید.

به مرحله ساخت فایل سیستم بروید

### مثال دوم: نصب آرچ در کنار درایوهای ویندوزی

با انتخاب دیسک مقصد (دیسکی که قرار است آرچ لینوکس بر روی آن نصب شود) cfdisk برای پارتیشن‌بندی دستی باز می‌شود.

با کلیدهای چپ و راست بین گزینه‌های پایین (Help, New, Print, ...) و با کلیدهای بالا و پایین بین پارتیشن‌های مختلف جابجا شوید.

```

cfdisk (util-linux 2.19.1)

Disk Drive: /dev/sda
Size: 126460362752 bytes, 126.4 GB
Heads: 255 Sectors per Track: 63 Cylinders: 15374

-----
Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot       Primary   ntfs         [System Reserved]  104.86*
sda2      Boot       Primary   ntfs         [System Reserved]  20865.62*
sda5      NC         Logical   ntfs         [System Reserved]  31458.33*
sda6      NC         Logical   ntfs         [System Reserved]  31456.29*
sda7      NC         Logical   Free Space   [System Reserved]  3.50*
-----
sda7      NC         Logical   ntfs         [System Reserved]  42570.74*
-----

[ Bootable ] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition_
  
```

تصویر ۳

این بار هارد دیسک خیلی نامرتبی داریم. بر روی sda1 و sda2 ویندوز هفت سوار شده و sda5 و sda6 مربوط به فایل‌هایی است که بر روی سیستم ذخیره کرده‌ایم. در این مرحله ما sda7 را برای نصب آرچ خالی کرده‌ایم، پس با حذف آن و تبدیل آن به پارتیشن‌های لینوکسی کار خود را آغاز می‌کنیم. ابتدا با کلید پایین آن را انتخاب می‌کنیم و سپس با Delete ظاهر شده آن را از روی هارد خود پاک می‌کنیم تا به Free Space یا همان فضایی خالی تبدیل شود.

**اخطار:** پاک کردن یک پارتیشن همانا و خداحافظی با اطلاعات موجود بر روی آن همانا. در هنگام انتخاب و پاک کردن پارتیشن مورد نظر نهایت دقت را انجام دهید چرا که اشتباه کوچک در اینجا مهلک به شمار می‌رود. به تازه‌کارها اکیداً توصیه می‌شود برای پاک کردن یک پارتیشن از برنامه‌هایی با رابط گرافیکی استفاده کنند.

```

cfdisk (util-linux 2.19.1)

Disk Drive: /dev/sda
Size: 126460362752 bytes, 126.4 GB
Heads: 255 Sectors per Track: 63 Cylinders: 15374

-----
Name        Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1        Boot      Primary   ntfs          [System Reserved]  104.86*
sda2        Primary   Primary   ntfs          20865.62*
sda5        NC        Logical   ntfs          31458.33*
sda6        NC        Logical   ntfs          31456.29*
-----
Pri/Log     Free Space  42574.24*
-----

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]

Create new partition from free space_

```

تصویر ۴

اکنون فضای خالی بوجود آمده را انتخاب می‌کنیم و از `New>Logical>Beginning` برای ساخت پارتیشن مربوط به ریشه اقدام می‌کنیم. اندازه این پارتیشن را پانزده هزار مگابایت انتخاب می‌کنیم. با انتخاب `Type` نوع این پارتیشن را `۸۳` (لینوکس) قرار می‌دهیم و پرچم راه‌اندازی را با انتخاب گزینه `Bootable` بر روی آن قرار می‌دهیم چرا که `/boot` در زیر مجموعه این پارتیشن قرار خواهد گرفت. به سراغ ساخت `/var` می‌رویم. مانند پارتیشن مربوط به ریشه از `New>Logical>Beginning` یک پارتیشن با حجم ده هزار مگابایت از نوع `۸۳` (لینوکس) ایجاد می‌کنیم. در قدم بعدی `swap` را با انتخاب فضای خالی و از طریق `New>Logical>Beginning` پارتیشنی به اندازه هزار مگابایت از نوع `۸۲` (`Linux swap / Solaris`) می‌سازیم. و در نهایت پارتیشن مربوط به خانه را از `New>Logical>Beginning` به اندازه فضای خالی باقی‌مانده و از نوع `۸۳` (لینوکس) ایجاد می‌کنیم. `sda7` را برای ریشه، `sda8` را برای `/var`، همچنین `sda9` را برای `swap` و `sda10` را برای خانه ایجاد کرده‌ایم (تصویر ۵).

```

cfdisk (util-linux 2.19.1)

Disk Drive: /dev/sda
Size: 126460362752 bytes, 126.4 GB
Heads: 255 Sectors per Track: 63 Cylinders: 15374

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1      Boot      Primary   ntfs         [System Reserved]  104.86*
sda2      Primary   Primary   ntfs         20865.62*
sda5      NC        Logical   ntfs         31458.33*
sda6      NC        Logical   ntfs         31456.29*
sda7      Boot      Logical   Linux        15002.92*
sda8      Logical   Logical   Linux        10001.95*
sda9      Logical   Linux swap / Solaris  1003.49*
sda10     Logical   Linux        16565.90*

[ Bootable ] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition_
  
```

تصویر ۵

نام پارتیشن‌های ایجاد شده و مقصد آن‌ها را برای اتصال در مرحله بعدی در گوشه‌ای یادداشت می‌کنیم.

با زدن کلیدهای چپ و راست گزینه Write را انتخاب می‌کنیم و برای تأیید yes را تایپ می‌کنیم و Enter را می‌زنیم. متوجه باشید که انجام این عمل اطلاعات موجود بر روی دیسک شما را با اطلاعات جدید بازنویسی می‌کند. گزینه Quit را برای خروج از برنامه انتخاب می‌کنیم. در ادامه Done را برای تأیید می‌زنیم.

تذکر: بعد از آخرین تغییرات هسته لینوکس که شامل ماژول‌های libata و PATA می‌شود تمامی درایوهای SATA، IDE و SCSI به عنوان sdx شناخته می‌شوند و نمایش اینگونه آن‌ها طبیعی است و جای نگرانی نیست.

نکته: در صورتی که ترجیح می‌دهید، دوباره مراحل را انجام دهید و از تنظیماتی که اعمال کرده‌اید ناراضی هستید کافیسیت با Quit از برنامه بدون این که Write را انجام دهید خارج شوید و cfdisk را مجدداً اجرا کنید.

به مرحله ساخت فایل سیستم بروید.

## پارتیشن بندی با gdisk

(یک هارد ۴۰ گیگابایتی)

برنامه را با دستور زیر بر روی هارد دیسک اول (sda) اجرا کنید:

```
# gdisk /dev/sda
```

### پاک کردن دیسک:

بعد از وارد کردن دستور فوق می‌توانید با تایپ علامت سؤال (?) گزینه‌های در دسترس را مشاهده کنید. با زدن دستور 0 کلیه اطلاعات موجود پاک‌شده و براساس جدول پارتیشن GUID هارددیسک شما بازنویسی می‌شود.

Command (? for help): o (overwrites existing partitions)

This option deletes all partitions and creates a new protective MBR. Proceed? (Y/N): y (yes)

### ساخت پارتیشن راه‌اندازی:

ابتدا n را برای ساخت یک پارتیشن جدید تایپ کنید. سپس شماره پارتیشن را وارد کنید (از ۱ تا ۱۲۸). سکتور اول را مشخص کنید (زدن Enter برای مقدار پیش‌فرض). سکتور آخر را با توجه به اندازه پارتیشن تعیین کنید. کد Hex را وارد کنید (ef02 برای پارتیشن بوت بایوس). برای دیدن مابقی کدها کلید L را فشار دهید.

Command (? for help): n (new)

Partition number (1-128, default 1): 1

First sector (34-83886046, default = 2048) or {+}size{KMGTP}: press enter to use the default

Last sector (2048-83886046, default = 83886046) or {+}size{KMGTP}: add size of partition (I use +3MB)

Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, Enter = 8300): ef02

Changed type of partition to 'BIOS boot partition'

### ساخت پارتیشن Swap:

نکته:

- برای سیستم‌هایی با رم بیش از ۲ گیگ وجود این پارتیشن غیر لازم است.

• بهتر است به جای پارتیشن فایل Swap را برای بهره‌وری از امکانات Swap همان‌گونه که پیش‌تر اشاره شد ایجاد کنید.

N را برای ساخت یک پارتیشن جدید فشار دهید و به مانند پارتیشن قبلی مراحل را ادامه دهید. در نهایت در کد Hex عدد ۸۲۰۰ را وارد کنید.

Command (? for help): n

Partition number (2-128, default 2): 2

First sector (34-83886046, default = 8192) or {+}size{KMGTP}: press enter to use the default

Last sector (8192-83886046, default = 83886046) or {+}size{KMGTP}: put your swap size here (e.g. +2GB)

Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, enter = 8300): 8200

Changed type of partition to 'Linux swap'

### ساخت ریشه

N را برای ساخت پارتیشن جدید وارد کنید و مراحل را تا کد Hex به مانند پارتیشن‌های دیگر پی بگیرید. در قسمت کد Hex مقدار پیش‌فرض را با زدن Enter انتخاب کنید.

Command (? for help): n

Partition number (3-128, default 3): 3

First sector (34-83886046, default = 4202496) (this varies based on other partition sizes) or {+}size{KMGTP}: press enter for default

Last sector (4202496-83886046, default = 83886046 or {+}size{KMGTP}: press enter for default

Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, enter = 8300): press enter

Changed type of partition to 'Linux filesystem'

### مرحله آخر

W را برای نوشتن تغییرات انتخاب کنید.

Command (? for help): w (write)

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING PARTITIONS!!

Do you want to proceed? (Y/N): y

به مرحله ساخت فایل سیستم بروید.

## ساخت فایل سیستم

**اخطار:** ساخت فایل سیستم باعث پاک شدن همه اطلاعات در پارتیشن مورد نظر خواهد شد.

### gdisk

ساخت فایل سیستم برای پارتیشن‌هایی که در بالا توسط `gdisk` ایجاد کردیم بدین صورت انجام می‌پذیرد. ابتدا فایل سیستم `swap` را برای `sda2` که به این منظور آماده کرده بودیم می‌سازیم و آن را روشن می‌کنیم:

```
# mkswap /dev/sda2 # (تنها در صورتی که این پارتیشن را برای این منظور آماده کرده‌اید)
# swapon /dev/sda2
```

سپس فایل سیستم `ext4` را برای پارتیشن `sda3` که به منظور ریشه آماده کرده بودیم ایجاد می‌کنیم:

```
# mkfs.ext4 /dev/sda3
```

به همین ترتیب می‌توانیم توسط `mkfs.ext4` فایل سیستم `ext4` را برای پارتیشن‌های دیگر بسازیم.

### cfdisk

به سراغ مثال اول با چهار پارتیشن می‌رویم.

اکنون نوبت به ساخت فایل سیستم برای پارتیشن‌هایی که توسط `cfdisk` ایجاد کرده‌ایم، رسیده است. `sda1` را برای ریشه، `sda2` را برای `/var`، همچنین `sda3` را برای `swap` و `sda4` را برای خانه ایجاد کرده‌ایم (تصویر ۲).

برای ساخت فایل سیستم `ext4` در پارتیشن‌های `sda1` و `sda2` و `sda4` دستورات زیر را وارد کنید.

```
# mkfs.ext4 /dev/sda1
# mkfs.ext4 /dev/sda2
# mkfs.ext4 /dev/sda4
```

پارتیشن `swap` هم بدین صورت فرمت و در نهایت روشن می‌کنیم:

```
# mkswap /dev/sda3
# swapon /dev/sda3
```

## متصل کردن پارتیشن‌ها

هر پارتیشن با یک نام پسونددار شناخته می‌شود. به عنوان مثال sda1 نمایانگر اولین پارتیشن بر روی درایو اول است. برای نمایش طرح پارتیشن‌های درایو اول از دستور زیر استفاده کنید:

```
# lsblk /dev/sda
```

**نکته:** دستوراتی که در ادامه برای متصل کردن پارتیشن‌ها ذکر شده با توجه به مثال اول در این آموزش بوده و مقادیر آن با توجه به نوع پارتیشن‌بندی شما می‌تواند متفاوت باشد.

ترتیب متصل کردن پارتیشن‌ها مهم است در نتیجه لازم است که دقت عمل به خرج دهید. ابتدا بایستی که پارتیشن ریشه را در /mnt متصل کنید.

```
# mount /dev/sda1 /mnt
```

در ادامه پارتیشن خانه و هر پارتیشن دیگری که به منظور استفاده در آرچ ساخته‌اید را سوار کنید. برای این منظور ابتدا باید مسیر مورد نظر را با استفاده از mkdir در زیر شاخه /mnt بسازید:

```
# mkdir /mnt/home
# mount /dev/sda4 /mnt/home
# mkdir /mnt/var
# mount /dev/sda2 /mnt/home
```

**نکته:** بیش از یک پارتیشن را به یک نقطه متصل نکنید.

اگر شما مادربرد UEFI دارید پارتیشن آن را نیز متصل کنید:

```
# mkdir /mnt/boot/efi
# mount /dev/sdaX /mnt/boot/efi
```



## انتخاب آینه<sup>۱</sup>

قبل از شروع نصب شما بایستی لیست آینه‌ها را ویرایش و آینه(های) مورد نظر خود را از حالت کامنت برای فعال شدن قابلیت دریافت خارج کنید. یک کپی از این فایل توسط `pacstrap` بر روی سیستم جدید شما ساخته خواهد شد پس ارزش وقت گذاشتن را دارد.

```
# nano /etc/pacman.d/mirrorlist
##
## Arch Linux repository mirrorlist
## Sorted by mirror score from mirror status page
## Generated on 2012-MM-DD
##

Server = http://mirror.example.xyz/archlinux/$repo/os/$arch
```

بعد از انتخاب آینه می‌توانید علامت(های) `#` را از جلوی خط بردارید تا فعال شود.

- `Alt+6` یک خط سرور را برای شما کپی می‌کند.
- `PageUp` به سمت بالا اسکرول می‌کند.
- `Ctrl+U` خط سرور را در بالای لیست می‌چسباند.
- `Ctrl+X` شما را از ویرایشگر خارج می‌کند. در صورتی که آینه‌ای را فعال کرده باشید در این حالت برنامه از شما اجازه می‌گیرد که تغییرات را ذخیره کند یا خیر. برای ذخیره کردن `Y` و سپس `Enter` را فشار دهید.

### ترفند:

- با استفاده از آینه‌ساز<sup>۲</sup> می‌توانید لیست از آینه‌های به‌روز را به دست آورید. آینه‌های HTTP از FTP سریع‌تر هستند. آینه‌های FTP برای دریافت هر بسته بایستی سیگنالی را ارسال کنند و این کار باعث یک وقفه کوچک در میان دریافت‌ها می‌شود.
- وضعیت‌دان آینه<sup>۳</sup> به شما گزارش در مورد آینه‌ها می‌دهد و شما را از مشکلات احتمالی آینه‌ها آگاه می‌سازد.

Mirror ۱

<https://www.archlinux.org/mirrorlist> ۲

<https://archlinux.org/mirrors/status> ۳

**نکته:**

- هرگاه در آینده لیست آیینه‌ها را تغییر دادید به یاد داشته باشید که به پک‌من برای به‌روزرسانی لیست بسته‌ها با استفاده از دستور زیر فشار بیاورید. این کار شما را از دردسرهای احتمالی خلاصی می‌دهد:

**# pacman -Syy**

- اگر شما از مدیای قدیمی برای نصب استفاده می‌کنید احتمالاً لیست آیینه‌ها نیز منسوخ شده است و برای شما مشکلاتی را پدید خواهد آورد. بهتر است لیست جدیداً را دریافت و از سرورهای آن استفاده کنید.
- برخی از مسائلی که در انجمن آرچ لینوکس مطرح گردیده حاکی از مشکل شبکه و جلوگیری از پک‌من برای به‌روزرسانی/همگام‌سازی مخازن است. در این زمان بهتر است که برنامه دریافت کننده پک‌من را تعویض کنید.<sup>۲</sup> زمانی که در حال نصب آرچ در وی‌باکس هستید این مشکل با استفاده از Host interface به جای NAT در machine properties حل خواهد شد.

## نصب بیس سیستم

به سراغ نصب بیس سیستم با استفاده از اسکریپت pacstrap می‌رویم. سوئیچ `i` می‌تواند مورد استفاده قرار بگیرد تا برای هر بسته‌ای از گروه‌های `base` و `base-devel` که قرار است نصب شود درخواست اجازه صادر نگردد.

```
# pacstrap -i /mnt base base-devel
```

۱ <https://www.archlinux.org/mirrorlist/all>

۲ قسمت بهبود عمل‌کرد پک‌من را مطالعه کنید.

**نکته:** اگر پک‌من در تأیید بسته‌های شما با مشکل مواجه شد با استفاده از `cal` ساعت سیستم را بررسی کنید. اگر تاریخ سیستم اشتباه بود (به عنوان مثال سال ۲۰۱۰ را نمایش می‌داد) کلیدها متقاضی شده یا فاقد اعتبار به حساب می‌آیند و تأیید امضای بسته‌ها با مشکل مواجه می‌شود و نصب قطع خواهد شد. مطمئن شوید که ساعت سیستم درست است. این کار را به صورت دستی یا با استفاده از کلاینت `ntp` می‌توانید انجام دهید سپس دستور فوق را مجدداً اجرا کنید. صفحهٔ زمان<sup>۲</sup> در ویکی آرچ می‌تواند کمک خوبی برای رفع مشکل باشد.

- `base`: بسته‌های نرم‌افزاری از مخزن `[core]` برای فراهم کردن محیط پایه‌ای کمینه.
  - `base-devel`: ابزار اضافی مانند `make` و `automake` از مخزن `[core]`. کاربران تازه‌وارد بایستی که آن‌ها را نصب کنند چرا که در آینده به آن‌ها نیاز خواهند داشت. گروه `base-devel` برای نصب برنامه‌ها از مخزن کاربران آرچ لازم است.
- شما تا اینجا یک سیستم پایه‌ای آرچ لینوکس دارید. بقیه بسته‌ها را توسط خود پک‌من نصب می‌کنیم.

## ساخت `fstab`

فایل `fstab` را با دستور زیر بسازید. این دستور `UUID` را با توجه به امتیازهایی که نسبت به بقیهٔ راه‌ها دارد برای شناسایی فایل سیستم‌ها به کار می‌گیرد. اگر شما می‌خواهید که از برچسب (`label`) پارتیشن به جای عبارت یکتای آن (`uuid`) استفاده کنید گزینهٔ `U` را با `L` عوض کنید.

**نکته:** اگر با مشکلی در حین اجرای `genfstab` مواجه یا در حین نصب مسئله‌ای پیش آمد این دستور را مجدداً اجرا نکنید و فایل `fstab` ساخته شده را ویرایش کنید.

```
# genfstab -U -p /mnt | sed 's/rw,relatime,data=ordered/defaults,relatime/' >> /mnt/etc/fstab
# nano /mnt/etc/fstab
```

**اخطار:** فایل `fstab` بعد از ساخته شدن بایستی حتماً بررسی شود. به همین دلیل ما بعد از دستور ساخت، دستور `nano` را برای مشاهدهٔ فایل صادر کردیم. اگر شما پیش‌تر یک پارتیشن

۱ به قسمت همسان‌سازی ساعت NTP مراجعه کنید.

۲ <https://wiki.archlinux.org/index.php/Time>

سیستمی EFI ساخته‌اید در نتیجه genfstab به اشتباه گزینه‌هایی را به پارتیشن سیستمی EFI اضافه می‌کند. این کار، راه‌اندازی کامپیوتر شما را از درایو مورد نظر با اختلال مواجه می‌کند پس شما بایستی تمام گزینه‌ها را برای پارتیشن EFI به غیر از noatime بردارید. برای بقیه پارتیشن‌هایی که از آن استفاده می‌کنند مطمئن شوید که "codepage=cp437" را با "codepage=437" تعویض می‌کند در غیر این صورت در شروع مجدد بعدی سیستم اتصال با مشکل مواجه خواهد شد و سیستم به حالت ریکاوری می‌رود. این مشکل بایستی با لینوکس ۳.۸ رفع گردیده باشد.

**نکته:** تنها پارتیشن ریشه (/) برای فیلد آخر نیاز به عدد 1 دارد. بقیه پارتیشن‌ها بایستی مقادیر 0 یا 2 را داشته باشند!

## Chroot و پیکربندی پایه

اکنون نوبت به chroot زدن به سیستم تازه نصب شده آرج رسیده است:

```
# arch-chroot /mnt
```

**نکته:** از arch-chroot /mnt /bin/bash برای chroot زدن به یک شل‌بش استفاده کنید.

در این مرحله از نصب شما فایل‌های اصلی پیکربندی آرج لینوکس را دستکاری خواهید کرد. این کار می‌تواند فایل‌هایی که اصلاً ساخته نشده را بسازد و یا چنانچه مایل باشید مقادیر آن‌ها را تغییر دهد.

## محل (Locale)

محل توسط glibc و بقیه برنامه‌های locale-aware یا کتابخانه‌ها برای رندر کردن متون، نمایش مقادیر پول محلی، قالب تاریخ و زمان، الفبا و بقیه استانداردهای خاص-محلی مورد استفاده قرار می‌گیرد.

دو فایل locale.gen و locale.conf در اینجا نیاز به ویرایش دارند.

۱ برای اطلاعات بیشتر در مورد fstab این صفحه (<http://linuxreview.ir/1390/08/mount-partitions-by-uuid>) چگونه fstab را ویرایش کنیم؟ را ببینید.

- فایل locale.gen یک فایل خالیست (همه چیز در آن کامنت شده و در اصل خالی در نظر گرفته می‌شود). شما بایستی علامت(های) # را از جلوی خط یا خطوطی که می‌خواهید، برای فعال کردن آن‌ها بردارید.  
لطفاً از مدخل‌های UTF-8 استفاده کنید:

```
# nano /etc/locale.gen
```

```
en_US.UTF-8 UTF-8
fa_IR UTF-8
```

```
# locale-gen
```

- بعد از این کار با هر به‌روزرسانی glibc تمام محل‌های مشخص شده در /etc/locale.gen را می‌سازد.
- فایل locale.conf به صورت پیش‌فرض ساخته نشده است. تنها تنظیم LANG کافی به نظر می‌رسد:

```
# echo LANG=en_US.UTF-8 > /etc/locale.conf
# export LANG=en_US.UTF-8
```

**نکته:** اگر شما زبان‌های دیگری غیر از انگلیسی را در ابتدای نصب تنظیم کرده‌اید دستور فوق چیزی شبیه به این خواهد بود:

```
# echo LANG=de_DE.UTF-8 > /etc/locale.conf
# export LANG=de_DE.UTF-8
```

برای استفاده از متغیرهای LC\_\* ابتدا locale را برای مشاهده گزینه‌های موجود اجرا کنید.<sup>۱</sup>

**اخطار:** استفاده از متغیر LC\_ALL ابداً توصیه نمی‌شود چرا که همه چیز را به هم می‌ریزد.

## فونت و نقشه‌کلید<sup>۲</sup> کنسول

اگر شما در ابتدا یک نقشه‌کلید را تنظیم کرده‌اید اکنون آن را بارگزاری کنید چرا که محیط

<sup>۱</sup> [https://wiki.archlinux.org/index.php/Locale#Setting\\_system-wide\\_locale](https://wiki.archlinux.org/index.php/Locale#Setting_system-wide_locale)

<sup>۲</sup> keymap

متفاوت شده است:

```
# loadkeys de-latin1
# setfont Lat2-Terminus16
```

برای این که بعد از شروع مجدد باز هم در دسترس باشند `vconsole.conf` را ویرایش کنید:

```
# nano /etc/vconsole.conf
```

```
KEYMAP=de-latin1
FONT=Lat2-Terminus16
```

- KEYMAP – به یاد داشته باشید که این تنظیمات تنها برای TTYهای شما معتبر است و برای هیچ مدیر پنجره گرافیکی یا Xorg اعتباری ندارد.
  - FONT – فونت‌های دیگری در دسترس را می‌توانید در `/usr/share/kbd/consolefonts/README.Lat2-Terminus16` مشاهده کنید (شامل ۱۱۰ ست زبانی).
  - گزینه‌های ممکن `FONT_MAP` – تعیین کننده نقشه کنسول برای بارگزاری در راه‌اندازی. این قسمت را خالی بگذارید یا پاک کنید.
- صفحه `man vconsole.conf` اطلاعات بیشتری را در اختیار شما قرار خواهد داد.

## منطقه زمانی

مناطق زمانی را می‌توانید در دایرکتوری `<SubZone>/<Zone>/usr/share/zoneinfo/` پیدا کنید. برای دیدن `<Zone>`های در دسترس مسیر `/usr/share/zoneinfo/` را بررسی کنید:

```
# ls /usr/share/zoneinfo/
```

به همین صورت `<SubZone>` را نیز مشاهده کنید:

```
# ls /usr/share/zoneinfo/Europe
```

یک لینک نمادی از `/etc/localtime` به فایل منطقه زمانی خود `<SubZone>/<Zone>/usr/share/zoneinfo/` با این دستور بسازید:

```
# ln -s /usr/share/zoneinfo/<Zone>/<SubZone> /etc/localtime
```

به عنوان مثال:

```
# ln -s /usr/share/zoneinfo/Iran /etc/localtime
```

## ساعت سخت افزار

تنظیم کردن نامناسب ساعت سخت افزار با ساعت سیستم عامل باعث بوجود آمدن تغییرات زمانی می شود.

شما می توانید `/etc/adjtime` را به صورت خودکار با دستوراتی که در ادامه آمده بسازید:

### UTC (توصیه می شود)

نکته: استفاده از UTC برای ساعت سخت افزار به این معنی نیست که نرم افزار هم ساعت مشابه را به نمایش بگذارد.

```
# hwclock --systohc --utc
```

برای همسان سازی زمان از طریق اینترنت، NTP راه حل مناسبی به نظر می رسد.

### همسان سازی ساعت NTP

ترفند: بسته ntp با فایل پیکربندی `/etc/ntp.conf` نصب شده است و بدون پیکربندی خاصی بایستی کار کند.

### تنظیم ارتباط با سرورهای NTP

در فایل `/etc/ntp.conf` سرورهایی که قرار است ساعت سیستم شما از طریق آنها همگام سازی شود مشخص شده است:

```
# nano /etc/ntp.conf

server 0.pool.ntp.org iburst
server 1.pool.ntp.org iburst
```

```
server 2.pool.ntp.org iburst
server 3.pool.ntp.org iburst
```

گزینهٔ `iburst` توصیه می‌شود.

مثال زیر یک فایل پیکربندی مناسب را به نمایش می‌گذارد:

```
# nano /etc/ntp.conf
```

```
server 0.pool.ntp.org iburst
server 1.pool.ntp.org iburst
server 2.pool.ntp.org iburst
server 3.pool.ntp.org iburst

restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery

restrict 127.0.0.1
restrict -6 ::1

driftfile /var/lib/ntp/ntp.drift
logfile /var/log/ntp.log
```

## همسان‌سازی ساعت

برای همسان‌سازی ساعت سیستم بدون اجرای دیمون کافیسست دستور زیر را صادر کنید:

```
# ntpd -qg
```

بعد از این کار بهتر است که ساعت سخت‌افزار خود را نیز تنظیم کنید تا از مشکلات بعد از شروع مجدد خلاصی یابید:

```
# hwclock -w
```

برای همسان‌سازی ساعت در زمان شروع به کار سیستم می‌توانید از روش زیر استفاده کنید. به یاد داشته باشید که این کار برای سیستم‌هایی که به ندرت شروع مجدد می‌شوند مناسب نمی‌باشد چرا که ساعت تا زمانی که سیستم ری‌استارت نشود تنظیم مجدد نمی‌گردد و ممکن است اختلاف زمانی پدید آید. از طرفی استفاده از `ntpd -qg` به عنوان یک `cron` هم پیشنهاد نمی‌شود چرا که ممکن است در کارهای برنامه‌های در حال اجرا اختلال ایجاد کند.

برای همسان‌سازی ساعت سیستم بعد از هر راه‌اندازی خط زیر را به فایل `/etc/rc.local` اضافه



کنید:

```
# nano /etc/rc.local
```

```
ntpd -qg &
```

همچنین مطمئن شوید که hwclock به آرایه‌های دیمون‌ها اضافه شده است<sup>۱</sup>.

### Localtime (توصیه نمی‌شود)

**اخطار:** استفاده از localtime می‌تواند شما را به سمت یک سری باگ شناخته شده غیرقابل تعمیر هدایت کند. به هر حال برنامه‌ای جهت عدم ساپورت localtime در حال اجرا نیست.

```
# hwclock --systohc --localtime
```

اگر شما قصد راه‌اندازی دوگانه<sup>۲</sup> ویندوز در کنار آرچ لینوکس را دارید در نتیجه:

- توصیه می‌شود. آرچ لینوکس و ویندوز را مجبور به استفاده از UTC کنید. فقط یک تعمیر ساده ریجیستری ویندوز نیاز است<sup>۳</sup>. همچنین مطمئن شوید که ویندوز همسان‌سازی زمان را انجام نمی‌دهد چرا که ساعت سخت‌افزار به localtime برگشت داده می‌شود و ساعت سیستم به هم خواهد ریخت.
- توصیه نمی‌شود: تمام سرویس‌های مربوط به زمان مانند NTPd را غیرفعال کنید و آرچ را در حالت localtime قرار دهید. کار تنظیم ساعت را به ویندوز بسپارید؛ با این کار شما بایستی حداقل ۲ بار در سال (بهار و پاییز) برای تصحیح ساعت به ویندوز وارد شوید.

### ماژول‌های هسته

**ترفند:** اینجا فقط یک مثال ذکر می‌کنیم در نتیجه لازم نیست که شما آن را انجام دهید. تمام ماژول‌های مورد نیاز به صورت خودکار توسط udev بارگیری می‌شوند بنابراین شما به ندرت نیاز به اضافه کردن چیزی در اینجا خواهید داشت. تنها ماژول‌هایی را اضافه کنید که مطمئن هستید باید

۱ به قسمت مدیریت دیمون‌ها مراجعه کنید.

۲ Dual boot

۳ [https://wiki.archlinux.org/index.php/Time#UTC\\_in\\_Windows](https://wiki.archlinux.org/index.php/Time#UTC_in_Windows) را ببینید.

اضافه شوند.

برای بارگیری ماژول‌های هسته در حین راه‌اندازی یک فایل `*.conf` در `/etc/modules-load.d/` با نام برنامه‌ای که از آن‌ها استفاده می‌کند، قرار دهید.

```
# nano /etc/modules-load.d/virtio-net.conf
```

```
# Load 'virtio-net.ko' at boot.
```

```
virtio-net
```

اگر ماژول‌های بیشتری برای بارگیری در یک `*.conf` وجود دارد می‌توانید آن‌ها را با یک خط خالی از هم جدا کنید.

خط‌های خالی یا خط‌هایی که با `#` و یا `;` شروع می‌شوند نادیده گرفته خواهند شد.

## نام میزبان (Hostname)

اسم میزبان را بنا به سلیقه خودتان انتخاب کنید (مثلاً `arch`).

```
# echo myhostname > /etc/hostname
```

نکته: نیازی به ویرایش `/etc/hosts` نیست. بسته `nss-myhostname` زحمت آن را خواهد کشید و در تمام سیستم‌ها به صورت پیش‌فرض نصب شده است.

## پیکربندی شبکه

در اینجا لازم است که مجدداً شبکه را پیکربندی کنید تا در سیستم جدیدی که نصب کرده‌اید بدون شبکه نباشید. این کار به مانند روشی که در ابتدای کتاب گفته شد است مگر در آنجایی که برقراری شبکه را بایستی در هنگام راه‌اندازی سیستم فعال کنیم.

نکته: برای اطلاعات جامع‌تر می‌توانید پیکربندی شبکه<sup>۱</sup> و تنظیمات وایرلس<sup>۲</sup> را مشاهده کنید.

## باسیم

### Dynamic IP

اگر شما تنها از یک ارتباط شبکه با سیستم استفاده می‌کنید در نتیجه نیاز به سرویس مدیریت شبکه نخواهید داشت و می‌توانید سرویس dhcpd را در جایی که <interface> رابط شماست به راحتی فعال کنید.

```
# systemctl enable dhcpd@<interface>.service
```

همچنین می‌توانید از netcfg برای این کار استفاده کنید. بسته ifplugd را که مورد نیاز net-auto-wired است را نصب کنید.

```
# pacman -S ifplugd
```

dhcp را تنظیم و net-auto-wired را فعال کنید:

```
# cd /etc/network.d
# ln -s examples/ethernet-dhcp .
# systemctl enable net-auto-wired.service
```

### Static IP

Ifplugd را نصب کنید:

```
# pacman -S ifplugd
```

پروفایل نمونه را از /etc/network.d/examples به /etc/network.d/ کپی کنید:

```
# cd /etc/network.d
# cp examples/ethernet-static .
```

پروفایل را با توجه به نیاز خود ویرایش کنید:

```
# nano ethernet-static
```

اطلاعات بیشتری را در اخطار بالا در مورد نام آن‌ها پیدا کنید. اگر `eth0` رابط شما نبود می‌توانید

```
WIRED_INTERFACE="<interface>"
```

سرویس `net-auto-wired` را فعال کنید:

```
# systemctl enable net-auto-wired.service
```

## بی‌سیم

شما در این حالت نیاز به نصب برنامه‌های بیشتری برای پیکربندی پروفایل شبکه بی‌سیم خود توسط `netcfg` دارید. `NetworkManager` و `Wicd` هم جایگزین‌های محبوبی برای این کار می‌باشند.

- بسته‌های مورد نیاز را نصب کنید:

```
# pacman -S wireless_tools wpa_supplicant wpa_actiond dialog
```

اگر وایرلس شما نیاز به سفت‌افزار دارد (همانگونه که در ابتدای کتاب اشاره شد) بایستی بسته مورد نیاز که شامل سفت‌افزار شما بشود را نصب کنید. به عنوان مثال:

```
# pacman -S zd1211-firmware
```

- بعد از تمام کردن نصب و شروع مجدد سیستم شما می‌توانید با استفاده از `wifi-menu <interface>` که یک فایل پروفایل در `/etc/network.d` با نام `SSID` بوجود می‌آورد به شبکه متصل شوید (`<interface>` نام رابط چیپست وایرلس شماست). همچنین نمونه‌هایی در `/etc/network.d/examples/` برای پیکربندی دستی وجود دارد.

```
# wifi-menu <interface>
```

**اخطار:** اگر شما از `wifi-menu` استفاده می‌کنید، این کار بایستی بعد از زمانی که سیستم را شروع مجدد کردید و دیگر در حالت `chroot` نیستید انجام بدهید. کاری که با این دستور انجام می‌گیرد با زمانی که در حالت خارج از `chroot` می‌شود متفاوت است. به عنوان راه جایگزین شما می‌توانید پروفایل شبکه را به صورت دستی با استفاده از

نمونه‌های قبلی که اشاره شد تنظیم کنید و در کل نگران استفاده از wifi-menu نباشید.

- سرویس net-auto-wireless را فعال کنید:

```
# systemctl enable net-auto-wireless.service
```

**نکته:** همچنین Netcfg می‌تواند net-auto-wired را برای استفاده در رابطه با net-auto-wireless فراهم نماید.

- مطمئن شوید که رابط بی‌سیم درست را در `/etc/conf.d/netcfg` تنظیم کرده‌اید. به عنوان مثال `wlp3s0`:

```
# nano /etc/conf.d/netcfg
```

```
WIRELESS_INTERFACE="wlp3s0"
```

همچنین امکان تعیین یک لیست از نمایه‌های شبکه که قرار است به صورت خودکار متصل شود با استفاده از متغیر `AUTO_PROFILES` در `/etc/conf.d/netcfg` وجود دارد. اگر `AUTO_PROFILES` تنظیم نشده است تمام شبکه‌های شناخته شده امتحان خواهند شد.

## پیکربندی پک‌من

پیش‌تر به طور مفصل در مورد پک‌من و پیکربندی آن مطالبی گفته شد. در اینجا لازم است که فایل پیکربندی را باز و مطابق میل خود پیکربندی لازم را انجام دهید:

```
# nano /etc/pacman.conf
```

برای اطلاعات بیشتر به بخش `/etc/pacman.conf` مراجعه کنید.

## ساخت initial ramdisk environment

ترفند: اغلب کاربران از این مرحله پرش می‌کنند و حالت پیش‌فرضی که در `mkinitcpio.conf` آماده شده را استفاده می‌کنند. ایمج `initramfs` (از پوشه `/boot`) بر اساس این فایل از قبل و در زمانی که بسته `linux` توسط `pacstrap` نصب شده بود ساخته شده است.

در اینجا شما بایستی `hook`های مناسب را اگر آرچ را بر روی یک فلش مموری نصب کرده‌اید، یا از `RAID`, `LVM` استفاده می‌کنید و یا `usr` بر روی یک پارتیشن جدا نصب شده است، تنظیم کنید. فایل `etc/mkinitcpio.conf` را بر اساس نیاز خود ویرایش و ایمج `initramfs` را مجدداً توسط دستور زیر بسازید:

```
# mkinitcpio -p linux
```

نکته: نصب `Arch VPS` بر روی `QEMU` (به عنوان مثال استفاده از `virt-manager`) احتمالاً به ماژول `virtio` در `mkinitcpio.conf` برای فعال کردن راه‌اندازی نیاز خواهد داشت:

```
# nano /etc/mkinitcpio.conf
```

```
MODULES="virtio virtio_blk virtio_pci virtio_net"
```

## تنظیم پسورد ریشه و اضافه کردن کاربر معمولی

با فرمان زیر برای کاربر ریشه کلمه عبوری تعیین کنید:

```
# passwd
```

اخطار: گنو/لینوکس یک سیستم‌عامل چندکاربره است. شما بایستی همه کارهای خود را با استفاده از حساب کاربری ریشه انجام دهید چرا که می‌تواند بسیار خطرناک باشد. اکانت ریشه تنها بایستی برای کارهای مدیریتی مورد استفاده قرار بگیرد.

اینک یک کاربر معمولی را اضافه کنید. مثلاً ما در اینجا کاربر `archie` را اضافه می‌کنیم:

```
# useradd -m -g users -G wheel -s /bin/bash archie
# passwd archie
```

اگر می‌خواهید کاربر را مجدداً اضافه کنید و کاربر قبلی را حذف نمایید از `userdel` استفاده کنید. در `userdel` چنانچه گزینه `-f` را بکار بگیرید دایرکتوری خانگی کاربر به همراه محتویات آن شامل تنظیمات کاربری نیز حذف خواهد شد.

```
# userdel -r archie
```

اطلاعات بیشتر را می‌توانید از صفحه کاربرها و گروه‌ها مشاهده کنید.

## نصب یک راه‌انداز (Boot Loader)

### مادربوردهای بایوس

برای سیستم‌های بایوس سه مدل راه‌انداز در دسترس می‌باشد. `Syslinux`، `GRUB` و `LILO`. در این مرحله انتخاب این که کدام وظیفه راه‌اندازی سیستم شما را بر عهده بگیرد بر عهده خود شماست. در اینجا نصب `Syslinux` و `GRUB` آموزش داده خواهد شد.

- `Syslinux` در حال حاضر تنها فایل‌هایی را از پارتیشنی که در آن (`Syslinux`) نصب شده است می‌تواند بارگیری کند. فایل پیکربندی آن ساده و قابل درک است.
- `GRUB`: یک برنامهٔ بال و پر دار که پشتیبانی بهتری از حالات مختلف دارد. فایل پیکربندی آن شباهت زیادی به یک زبان اسکریپت‌نویسی دارد که ممکن است برای تازه‌کارها مشکل باشد. برای این دسته از افراد پیشنهاد می‌شود که این فایل را به صورت خودکار بسازند.

### Syslinux

بستهٔ `Syslinux` را نصب کنید و سپس اسکریپت `syslinux-install_update` را برای نصب خودکار فایل‌ها بکار بگیرید (`-i`)، پارتیشن فعال را با پرچم راه‌اندازی نشانه‌گذاری کنید (`-a`) و کدهای راه‌اندازی `MBR` را نصب کنید (`-m`).

```
# pacman -S syslinux
# syslinux-install_update -i -a -m
```

با پیکربندی `syslinux.cfg` آدرس صحیح ریشه را مشخص کنید. این مرحله حیاتی است و در صورتی که شما آدرس اشتباهی انتخاب کنید آرچ لینوکس راه اندازی نخواهد شد. در اینجا `/dev/sda3` را به پارتیشن ریشه که در مثال‌های بالا آدرس آن را مشخص کردیم تغییر دهید. همین کار را برای مدخل `fallback` انجام دهید:

```
# nano /boot/syslinux/syslinux.cfg
```

```
...
LABEL arch
...
APPEND root=/dev/sda3 ro
...
```

## GRUB

بسته `grub-bios` را نصب کنید سپس دستور `grub-install` را اجرا کنید:

**نکته:** `/dev/sda` را به درایوی که آرچ (پارتیشن مربوط به راه اندازی [جایی که `/boot` واقع شده است]) بر روی آن نصب شده تغییر دهید. به یاد داشته باشید که شماره پارتیشن را مشخص نکنید. به عبارت دیگر آن را به صورت `sdaX` (به همراه شماره پارتیشن که در اینجا X است) ننویسید.

**نکته:** برای درایوها با پارتیشن‌های GPT در مادربردهای بایوس بایستی یک پارتیشن بوت بایوس ۲ مگابایتی داشته باشید.

```
# pacman -S grub-bios
# grub-install --target=i386-pc --recheck /dev/sda
# cp /usr/share/locale/en@quot/LC_MESSAGES/grub.mo /boot/grub/locale/en.mo
```

با وجود این که ساخت دستی `grub.cfg` کار بسیار خوبی است اما برای تازه‌کارها پیشنهاد می‌شود که آن را به صورت خودکار آماده کنند.

**ترفند:** برای این که به صورت خودکار سیستم‌عامل‌های دیگر نصب شده بر روی کامپیوتر شما شناسایی و اضافه شوند ابتدا بسته `os-prober` را توسط دستور زیر نصب کنید.

```
# pacman -S os-prober
```



```
# grub-mkconfig -o /boot/grub/grub.cfg
```

## مادربردهای UEFI

در اینجا لازم است که درایو شما به صورت GPT پارتیشن بندی شده باشد و یک پارتیشن (۵۱۲ مگابایتی یا بیشتر، FAT32 و نوع EF00) بایستی حاضر و به `/boot/efi` متصل شده باشد.

**نکته:** Syslinux از UEFI پشتیبانی نمی‌کند.

## EFISTUB

هسته لینوکس می‌تواند با استفاده از EFISTUB به عنوان راه‌انداز عمل کند. این راه توصیه شده برای راه‌اندازی توسط توسعه دهندگان است. در ادامه تنظیمات rEFInd را برای فراهم کردن یک منوی هسته EFISTUB انجام می‌دهیم. شما همچنین می‌توانید از gummiboot به جای rEFInd استفاده کنید. هر دوی آن‌ها می‌توانند راه‌اندازی UEFI ویندوز را شناسایی و راه‌اندازی چندگانه را فعال کنند.

۱. راه‌اندازی را در مد UEFI انجام داده و ماژول هسته efivars را قبل از chroot بارگیری کنید:

```
# modprobe efivars # before chrooting
```

۲. پارتیشن UEFISYS را به `/mnt/boot/efi` متصل کنید، سپس chroot زده و فایل‌های `initramfs` و هسته را همان‌گونه که در ادامه توضیح داده شده کپی کنید:

- مسیر `/boot/efi/EFI/arch/` را بسازید.
- `/boot/vmlinuz-linux` را به `/boot/efi/EFI/arch/vmlinuz-arch.efi` کپی کنید. فایل با پسوند `.efi` فایل‌های بسیار حیاتی است. توجه کنید که با اسم `vmlinuz` شناخته می‌شود نه با اسم `vmlinuX`.
- `/boot/initramfs-linux.img` را در `/boot/efi/EFI/arch/initramfs-arch.img` کپی کنید.
- `/boot/initramfs-linux-fallback.img` را به `/boot/efi/EFI/arch/initramfs-arch-fallback.img` نیز کپی کنید.

هر زمان هسته و فایل‌های `initramfs` در `/boot` ارتقاء پیدا کنند بایستی آن‌ها را در

۳. در این راهنما شما نحوه تنظیم رابط گرافیکی راه‌انداز که rEFInd نامیده می‌شود را آموزش خواهید دید. بسته‌های زیر را نصب کنید:

```
# pacman -S refind-efi efibootmgr
```

۴. rEFInd را در پارتیشن UEFISYS نصب کنید:

```
# mkdir -p /boot/efi/EFI/refind
# cp /usr/lib/refind/refind_x64.efi /boot/efi/EFI/refind/refind_x64.efi
# cp /usr/lib/refind/config/refind.conf /boot/efi/EFI/refind/refind.conf
# cp -r /usr/share/refind/icons /boot/efi/EFI/refind/icons
```

۵. فایل refind\_linux.conf را با پارامترهای مناسب هسته بسازید:

```
# nano /boot/efi/EFI/arch/refind_linux.conf
```

```
"Boot to X" "root=/dev/sdaX ro rootfstype=ext4 systemd.unit=graphical.target"
"Boot to console" "root=/dev/sdaX ro rootfstype=ext4 systemd.unit=multi-user.target"
```

نکته: در دایرکتوری /boot/efi/EFI/arch/ refind\_linux.conf جای که initramfs و هسته در مرحله دوم کپی شده‌اند رونوشت شده است.

نکته: در refind\_linux.conf عبارت sdaX به فایل سیستم ریشه شما برمی‌گردد و اگر شما پارتیشن ریشه را جدا درست کرده باشید باز هم به فایل سیستم ریشه برمی‌گردد، نه به پارتیشن ریشه.

۶. rEFInd را به منوی راه‌اندازی UEFI با استفاده از efibootmgr اضافه کنید.

اخطار: استفاده از efibootmgr در Apple Mac ها ممکن است برای سفت‌افزار مشکل ایجاد کرده و شما را مجبور به فلش کردن دوبارهٔ رام مادربرد کند. در مک‌ها از mactel-boot<sup>۳</sup> استفاده کنید.

۱ [https://wiki.archlinux.org/index.php/UEFI\\_Bootloaders#Sync\\_EFISTUB\\_Kernel\\_in\\_UEFISYS\\_partition\\_using\\_Systemd](https://wiki.archlinux.org/index.php/UEFI_Bootloaders#Sync_EFISTUB_Kernel_in_UEFISYS_partition_using_Systemd)

۲ [https://wiki.archlinux.org/index.php/UEFI\\_Bootloaders#Sync\\_EFISTUB\\_Kernel\\_in\\_UEFISYS\\_partition\\_using\\_Incron](https://wiki.archlinux.org/index.php/UEFI_Bootloaders#Sync_EFISTUB_Kernel_in_UEFISYS_partition_using_Incron)

۳ <https://aur.archlinux.org/packages/mactel-boot>

```
# efibootmgr -c -g -d /dev/sdX -p Y -w -L "rEFInd" -l "\EFI\refind\refind_x64.efi"
```

**نکته:** در فرمان فوق X و Y به درایو و پارتیشن UEFISYS اشاره دارد. به عنوان مثال در `/dev/sdc5` مقدار X برابر با C و مقدار Y برابر با 5 است.

۷. (اختیاری) اگر در اینجا مدخل راه‌اندازی ساخته شده توسط `efibootmgr` عمل نکرد `refind_x64.efi` را به `/boot/efi/EFI/boot/boot_x64.efi` کپی کنید:

```
# cp -r /boot/efi/EFI/refind/* /boot/efi/EFI/boot/
# mv /boot/efi/EFI/boot/refind_x64.efi /boot/efi/EFI/boot/boot_x64.efi
```

## GRUB

**نکته:** در این حالت فرض می‌گیریم که شما یک سیستم ۳۲-بیت EFI (مانند مک‌های قبل از ۲۰۰۸) دارید. `Grub-efi-i386` را نصب و از `target=i386-efi` استفاده می‌کنیم.

```
# pacman -S grub-efi-x86_64 efibootmgr
# grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=arch_grub --recheck
# cp /usr/share/locale/en@quot/LC_MESSAGES/grub.mo /boot/grub/locale/en.mo
```

دستور بعد یک ورودی منو برای گراب در منوی راه‌اندازی UEFI می‌سازد. به هر حال در `efibootmgr` نسخه `grub-efi-x86_64` دو `grub-install` سعی در ساخت یک ورودی منو می‌کند بنابراین مورد نیاز نخواهد بود.

```
# efibootmgr -c -g -d /dev/sdX -p Y -w -L "Arch Linux (GRUB)" -l "\EFI\arch_grub\grubx64.efi"
```

در این مرحله با وجود این که می‌توانید `grub.cfg` را به صورت دستی بسازید بهتر است به صورت خودکار آن را ایجاد کنید:

**ترفند:** برای این که به صورت خودکار سیستم‌عامل‌های دیگر نصب شده بر روی کامپیوتر شما شناسایی و اضافه شوند ابتدا بسته `os-prober` را توسط دستور زیر نصب کنید.

```
# pacman -S os-prober
```

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

## پیاده کردن پارتیشن‌ها و شروع مجدد

از محیط chroot خارج شوید:

```
# exit
```

با توجه به پارتیشن‌هایی که در زیر مسیر /mnt متصل شده‌اند همه آن‌ها را پیاده کنید:

```
# umount /mnt/{boot,home,}
```

دستور شروع مجدد را برای خارج شدن و رفتن به محیط سیستم‌عامل نصب شده صادر کنید:

```
# reboot
```

ترفند: مطمئن شوید که رسانه نصب را از دستگاه خارج می‌کنید چرا که ممکن است به صورت ناخودآگاه مجدداً به محیط دیسک زنده هدایت شوید.

۱ ما برای mount از واژه سوار کردن یا متصل کردن و برای umount از واژه پیاده کردن در این کتاب استفاده می‌کنیم. پیاده کردن یک پارتیشن یعنی قطع اتصال پارتیشن به نقطه‌ای که قبلاً آن را متصل کرده بودیم.

# بخش سوم

## اضافات

تبریک می‌گوییم. شما با موفقیت آرچ لینوکس را نصب کردید و اکنون پذیرای خوش‌آمد گویی گرم ما باشید.

آرچ لینوکس شما در این مرحله یک محیط گنو/لینوکس کاربردیست و آماده است که طبق سلیقه شما چیده و پیکربندی شود. بیشتر افراد ترجیح می‌دهند که یک میزکار با صدا و گرافیک داشته باشند. شما چطور؟ در این قسمت از کتاب خواهید فهمید که چگونه محیط را مطابق میل خود بچینید.

### مدیر بسته (PACKAGE MANAGEMENT) ۲

با پک‌من آشنا شوید. بهترین رفیق شما در آرچ لینوکس بدون شک پک‌من است. شدیداً توصیه می‌شود که برای یادگیری بیشتر کار با آن صفحه man مربوط به آن را مطالعه کنید.

```
$ man pacman
```

Desktop ۱

۲ این قسمت شامل آموزش‌هایی که ممکن است در طول کتاب تکرار شده باشد، می‌شود.

## موارد استفاده پک‌من

مدیر بسته پک‌من یکی از نقاط قوت آرچ لینوکس محسوب می‌شود. هدف پک‌من مدیریت آسان بسته‌هاست چه در مخازن رسمی آرچ باشند و چه بسته‌های ساخته شده توسط خود کاربر.

فایل پیکربندی پک‌من در مسیر `/etc/pacman.conf` مستقر شده است. برای اطلاعات بیشتر `man pacman.conf` را فراخوانی کنید. به عنوان مثال شما می‌توانید در این فایل مشخص کنید که پک‌من برای دانلود بسته‌ها از برنامه خاصی استفاده کند یا نه و همچنین در هنگام به‌روزرسانی و یا نصب بسته‌ها از به‌روزرسانی یا نصب بسته یا گروهی از بسته‌ها پرش کند. همچنین در این فایل می‌توانید مخازن را اضافه یا کم کنید.

## نصب بسته

برای نصب یک بسته کفایت دستور زیر را صادر کنید. برای نصب دو یا چند بسته کفایت بین آن‌ها را با یک فاصله مشخص کنید:

```
# pacman -S package_name1 package_name2 ...
```

در برخی مواقع شما یک بسته را با نسخه‌های مختلف در مخازن مختلف دارید. در این هنگام برای این که به پک‌من بفهمانید که کدام بسته را نصب کند لازم است مخزن آن بسته را نیز مشخص کنید:

```
# pacman -S extra/package_name
# pacman -S testing/package_name
```

برای نصب بسته‌های یک گروه به عنوان مثال `gnome`:

```
# pacman -S gnome
```

این دستور تمام بسته‌هایی که به گروه گنوم متعلق هستند را نصب می‌کند. برای دیدن بسته‌های متعلق به یک گروه به عنوان مثال گنوم:

```
# pacman -Sg gnome
```

تذکر: بسته‌های گروهی موجود از صفحهٔ مربوط به آن قابل مشاهده‌اند. چنانچه بسته‌ای از قبل بر روی سیستم شما نصب شده باشد تا زمانی که گزینهٔ `-needed` را بکار نگیرید حتی اگر بسته به‌روز هم باشد باز توسط پک‌من مجدداً نصب خواهد شد.

## حذف بسته

برای حذف بسته بدون حذف پیش‌نیازهایش از دستور زیر استفاده کنید:

```
# pacman -R package_name
```

برای حذف بسته و پیش‌نیازهایی که توسط بسته‌های نصب شدهٔ دیگر مورد نیاز نیستند از دستور زیر استفاده کنید:

```
# pacman -Rs package_name
```

برای حذف بسته و پیش‌نیازهایش به علاوه تمام بسته‌های که به بستهٔ مقصد محتاج هستند از دستور زیر استفاده کنید.

**اخطار:** در بکارگیری این دستور دقت کنید چون می‌تواند به صورت بالقوه بسته‌های مورد نیاز را حذف کند.

```
# pacman -Rsc package_name
```

پک‌من فایل‌های مهم پیکربندی را زمانی که یک برنامه را حذف می‌کنید با نامی مشخص و پسوند `.pacsave` نگه داری می‌کند. برای جلوگیری از این کار پک‌من می‌توانید از گزینهٔ `-n` استفاده

کنید:

```
# pacman -Rn package_name
```

برای حذف یک بسته که توسط بسته دیگر مورد نیاز است بدون حذف بسته وابسته:

```
# pacman -Rdd package_name
```

**تذکر:** پک من فایل‌های پیکربندی که توسط خود برنامه‌ها ساخته شده را حذف نمی‌کند. برای دیدن این فایل‌های پوشه خانگی خود را کاوش کنید.

## ارتقاء بسته‌ها

پک من می‌تواند تمام بسته‌های موجود بر روی سیستم را با یک دستور به‌روز کند. این دستور پایگاه داده مخازن را همگان سازی می‌کند و سیستم را با بسته‌های به‌روز، به‌روزرسانی می‌کند:

```
# pacman -Syu
```

**اخطار:** با توجه به انتشار غلطان کاربر بایستی توجه داشته باشد به‌روز رسانی‌های سریع و پی در پی ممکن است پیامدهای غیرمنتظره‌ای را به همراه داشته باشد. به عبارت دیگر به‌روزرسانی سریع بعد از انتشار هر برنامه انتخاب هوشمندانه‌ای نیست و بهتر است شما در زمانی که وقت آزاد برای مقابله با مشکلاتی که به ندرت در حین به‌روزرسانی پدیدار می‌شود را در اختیار دارید اقدام به به‌روزرسانی کل سیستم ننمائید. قبل از ارتقاء سیستم بهتر است صفحه خانگی آرچ لینوکس را باز کنید و آخرین اخبار را مطالعه نمائید.

## دیگر گزینه‌های پک من

**نکته:** فایل‌های گزارش پک من در مسیر `/var/log/pacman.log` نگه داری می‌شوند.

پک من با `-Q` دیتابیس بسته‌های محلی را مورد کاوش قرار می‌دهد:



```
$ pacman -Q --help
```

و دیتابیس بسته‌های موجود در مخازن را با:

```
$ pacman -S --help
```

پک‌من می‌تواند بسته‌های موجود در دیتابیس را با توجه به نام و توضیحاتشان مورد جستجو قرار

دهد:

```
$ pacman -Ss string1 string2 ...
```

و همچنین جستجوی بسته‌های نصب شده:

```
$ pacman -Qs string1 string2 ...
```

برای نمایش اطلاعات گسترده یک بسته:

```
$ pacman -Si package_name
```

و بسته‌های نصب شده:

```
$ pacman -Qi package_name
```

گذاشتن دو `-i` لیست فایل‌های پشتیبان را به نمایش می‌گذارد:

```
$ pacman -Qii package_name
```

برای دریافت لیست فایل‌هایی که توسط یک بسته نصب شده:

```
$ pacman -Ql package_name
```

همچنین می‌توانید از دیتابیس برای تشخیص این که هر فایل توسط چه بسته‌ای نصب شده استفاده کنید:

```
$ pacman -Qo /path/to/a/file
```

برای دریافت لیست از بسته‌هایی که دیگر به عنوان پیش‌نیاز مورد استفاده قرار نمی‌گیرند (بی‌سرپرست):

```
$ pacman -Qdt
```

برای دریافت لیست درختی پیش‌نیازهای یک بسته:

```
$ pactree package_name
```

برای دریافت لیست تمام بسته‌های وابسته به یک بسته<sup>۱</sup>:

```
$ whoneeds package_name
```

به‌روزرسانی سیستم و نصب بسته یا بسته‌های خاص بعد از آن:

```
# pacman -Syu package_name1 package_name2 ...
```

دانلود بسته‌ها بدون نصب آن‌ها:

```
# pacman -Sw package_name
```

نصب بسته محلی (نه از مخازن) (این گزینه همچنین برای دانگريد برنامه‌ها از طريق بسته‌های موجود در کش یا هر محل دیگری مورد استفاده قرار می‌گیرد):

```
# pacman -U /path/to/package/package_name-version.pkg.tar.xz
```

۱ برای استفاده از این فرمان لازم است بسته Pkgtools را نصب کرده باشید.

نصب بسته از راه دور (نه از مخازن):

```
# pacman -U http://www.example.com/repo/example.pkg.tar.xz
```

پاک کردن بسته‌های موجود در کشِ پک‌من که در حال حاضر بر روی سیستم نصب نشده‌اند:

```
# pacman -Sc
```

پاک کردن کلیه بسته‌های موجود در کشِ پک‌من:

```
# pacman -Scc
```

**اخطار:** توجه داشته باشید با پاک کردن بسته‌های موجود در کش دیگر قادر به دانگرید نرم‌افزارها توسط آن‌ها نخواهید بود.

چنانچه پس از مدتی احساس کردید که پک‌من شما در زمان تشخیص پیش‌نیازها و آماده کردن اطلاعات کند عمل می‌کند دستور زیر را برای بهبود عمل کرد آن در ترمینال صادر کنید:

```
# pacman-optimize
```

برای اعمال فشار جهت نصب بسته می‌توانید از گزینه `-f` استفاده کنید (استفاده از این گزینه به هیچ عنوان پیشنهاد نمی‌شود):

```
# pacman -Sf package_name
```

ممکن است در برخی از مواقع در هنگام نصب برنامه‌ای با این خط `"file exists in filesystem"` مواجه شوید. در این هنگام با استفاده از `pacman -Qo` مطمئن شوید که فایل به برنامه دیگری متعلق نباشد (چنانچه بود گزارش باگ بدهید). سپس با استفاده از گزینه `-f` برای اعمال فشار جهت بازنویسی فایل مورد نظر اقدام به نصب بسته کنید. البته توصیه می‌شود فایلی که پک‌من به آن اشاره می‌کند (که از قبل وجود دارد) را تغییر نام دهید و بدون استفاده از گزینه `-f` مجدداً اقدام به نصب

بسته مورد نظر کنید.

چنانچه پک من در حین حذف، نصب یا به روزرسانی بسته‌ای با خطای "database write" مواجه شد بایستی بسته شکسته را اینگونه تعمیر کنید:

- به دیسک زنده آرچ بروید.
- فایل سیستم ریشه را متصل کنید.
- دیتابیس پک من را توسط pacman-Syy به روز کنید.
- بسته شکسته را با استفاده از `pacman -r /path/to/root-S package` نصب کنید.

## /etc/pacman.conf

هر بار که پک من درگیر کاری می‌شود اطلاعات داخل `/etc/pacman.conf` را می‌خواند. این فایل پیکربندی به قسمت‌ها و مخزن‌های مختلف تقسیم شده است. هر بخش مشخص کننده مخزن بسته‌هاست و پک من با توجه به این اطلاعات اقدام به جستجوی بسته‌ها از داخل مخازن می‌کند. یک قسمت استثناء در این فایل قرار دارد که مشخص کننده گزینه‌های مربوط به پک من می‌شود.

**تذکر:** این فایل به صورت پیش فرض به خوبی پیکربندی شده و نیاز به ایجاد تغییرات غیر لازم در آن در این مرحله ضروری نمی‌باشد اما بررسی آن همیشه توصیه می‌شود.

```
# nano /etc/pacman.conf
```

مخازن در انتهای فایل قرار گرفته‌اند و شما می‌توانید با برداشتن علامت # از ابتدای خطوط `Include =` و `[repository]` آن‌ها را فعال کنید.

**تذکر:** زمانی که یک مخزن را اضافه می‌کنید مطمئن شوید دو خط `Include =` و `[repository]` از حالت کامنت خارج شده‌اند. عدم این کار باعث برخورد با خطاهایی می‌شود که بسیار متداول است.

برای این که پک من به صورت مستمر از به روزرسانی بسته یا گروهی از بسته‌ها صرف نظر کند به ترتیب یکی از موارد زیر را در فایل تنظیمات مشخص کنید. در مثال زیر پک من از به روزرسانی بسته

linux و گروه gnome پرش می‌کند:

```
# nano /etc/pacman.conf
```

```
IgnorePkg=linux
IgnoreGroup=gnome
```

## مخازن بسته‌ها

یک مخزن نرم‌افزاری محلی است که بسته‌های نرم‌افزاری از آن گرفته و بر روی کامپیوتر نصب می‌شوند. نگاه‌دارندگان بسته‌های آرچ لینوکس (توسعه‌دهندگان و کاربران قابل اعتماد) یک سری از مخازن رسمی که شامل بسته‌های نرم‌افزاری ضروری، محبوب و عمومی که به سهولت از طریق پک‌من قابل نصبند را نگه‌داری می‌کنند. برای اطلاعات بیشتر از قبیل هدف هر مخزن مخازن رسمی را مورد مطالعه قرار دهید.

اغلب مردم از مخزن‌های [extra], [core], و [community] استفاده می‌کنند. اگر شما تمایل به اجرای ابزارهای ۳۲-بیت بر روی آرچ x86\_64 دارید می‌توانید مخزن [multilib] را با اضافه کردن خطوط زیر به انتهای فایل /etc/pacman.conf فعال کنید.

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```

## مخزن کاربران آرچ AUR

مخزن کاربران آرچ (AUR) شامل شاخه‌های پشتیبانی نشده که مستقیماً از طریق پک‌من قابل نصب نیستند می‌شود. [AUR unsupported] شامل بسته‌های باینری خاص و آماده شده نمی‌باشد. بلکه شامل هزاران اسکریپت PKGBUILD است که بسته‌ها را از کد منبع می‌سازد. این بسته‌ها ممکن است از طریق مخازن دیگر در دسترس نباشند. چنانچه بسته‌های [unsupported] به مقدار کافی رأی مثبت دریافت کنند به [community] AUR منتقل می‌شوند و کاربران قابل اعتماد می‌توانند آن‌ها را پذیرفته و نگه‌داری کنند.

تذکره: یک سری پوشش برای پک‌من وجود دارد (AUR Helpers) که می‌تواند برای شما امکان دسترسی مستقیم به AUR را فراهم کند.

برای نصب برنامه از این مخزن به قسمت yaourt مراجعه کنید.

## /etc/pacman.d/mirrorlist

این فایل مشخص کننده آینه‌ها و اولیت‌هایشان برای مخازن پک‌من است.

توسط یک ویرایشگر (مانند nano) فایل /etc/pacman.d/mirrorlist را باز کرده و سرورهای نزدیک به محل زندگی خود را با برداشتن علامت # ابتدای آن‌ها از حالت کامنت خارج کنید. سپس اقدام به یک به‌روزرسانی توسط دستور زیر نمائید:

```
# pacman -Syy
```

گذاشتن دو عدد refresh-- یا -y به پک‌من برای به‌روزرسانی کلیه لیست بسته‌ها فشار وارد می‌کند و ما را از به‌روز بودن آن مطمئن می‌کند. زمانی که یک آینه‌ای تغییر کرد pacman -Syy را اجرا کنید تا از هرگونه مشکل احتمالی در امان باشید.

## بهبود عمل کرد پک‌من

### بهبود سرعت دسترسی به دیتابیس

پک‌من تمام اطلاعات بسته‌ها را در فایل‌های کوچکی نگه می‌دارد، یک فایل برای هر بسته. با بهبود سرعت دسترسی، زمانی را که این وظیفه برای جستجو در بین بسته‌ها انجام می‌دهد را کاهش دهید. کفایت که هر از چندگاهی دستور زیر را با مجوز ریشه اجرا کنید:

```
# pacman-optimize
```

این کار تمام فایل‌های کوچک را در کنار هم و در یک فضای نزدیک قرار می‌دهد در نتیجه هارد دیسک شما برای دسترسی به آن‌ها کار سختی را در پیش رو نخواهد داشت. این کار بی‌خطر است ولی بستگی به میزان فضای خالی بر روی فایل سیستم شما دارد. کار دیگری که پیشنهاد می‌شود انجام دهید پاک کردن بسته‌های اضافیست. با دستور زیر بسته‌هایی که بر روی سیستم شما نصب نیستند را پاک خواهید کرد. شاید نسخه جدیدتری از این بسته‌ها وجود داشته باشد و یا شاید شما آن‌ها را دریافت کرده باشید و در ادامه آن‌ها را حذف کرده باشید اما همچنان بسته‌ها باقی مانده باشند:

```
# pacman -Sc && pacman-optimize
```

برای یک تمیزکاری مفصل از `pacman -Sc` استفاده کنید. توجه داشته باشید که کلیه بسته‌های موجود در کش پک‌من<sup>۱</sup> حذف خواهد شد و قابلیت نصب مجدد بدون دریافت و دانگرید آسان<sup>۲</sup> برنامه‌ها از شما گرفته می‌شود.

### بهبود سرعت دریافت

اگر سرعت شما به شدت کاهش یافته است مطمئن شوید که از آینه‌های دیگری غیر از `ftp.archlinux.org` استفاده می‌کنید.

سرعت دریافت در پک‌من می‌تواند با استفاده از برنامه دیگری غیر از دریافت‌کننده داخلی آن افزایش یابد. در هر صورت قبل از هر تغییری مطمئن شوید که از آخرین نسخه آن استفاده می‌کنید:

```
# pacman -Syu
```

### استفاده از wget

برای استفاده از `wget` ابتدا آن را نصب کنید:

```
# pacman -S wget
```

سپس با اضافه کردن خط زیر به قسمت `[options]` در فایل `/etc/pacman.conf` آن را فعال کنید<sup>۳</sup>:

```
# nano /etc/pacman.conf
```

```
XferCommand = /usr/bin/wget -c --passive-ftp -c %u
```

- ۱ Pacman Cache حافظه نهانیست که فایل‌های دریافتی ذخیره می‌شود. شما با استفاده از این فایل‌ها می‌توانید بسته‌ای را بدون نیاز به دریافت دوباره نصب کنید و یا با آن به نسخه‌های قدیمی‌تر از برنامه دانگرید کنید.
- ۲ دانگرید برنامه به عملی اطلاق می‌شود که شما را به نسخه قبل‌تر از نسخه نصبی، انتقال می‌دهد. به عنوان مثال زمانی که شما با نسخه جدیدی از برنامه‌ای مشکلی پیدا می‌کنید که در نسخه قبلی موجود نبود، می‌توانید با استفاده از بسته‌های نسخه قبلی موجود در کش پک‌من برنامه را به نسخه قبل بازگردانید. به دانگرید برنامه‌ها مراجعه کنید.
- ۳ احتمالاً این خط در قسمت `[options]` موجود است و شما کافیست که آن را از حالت کامنت با برداشتن علامت `#` از ابتدای آن خط فعال کنید.

اطلاعات بیشتر در زمینه بهینه‌سازی پک‌من را می‌توانید از صفحهٔ ویکی آن دریافت کنید<sup>۴</sup>.

## مدیر سیستم (systemd)

آرچ‌لینوکس از systemd به عنوان init استفاده می‌کند. Systemd وظیفهٔ مدیریت سیستم و سرویس‌ها را به عهده دارد و به شما توصیه می‌شود که با اصول اولیهٔ آن حتماً آشنایی پیدا کنید. دستور systemctl روزه‌ای به درجهٔ systemd در آرچ‌لینوکس به حساب می‌آید. برای اطلاعات بیشتر دستور `man 1 systemctl` را در ترمینال صادر کنید.

ترفند: شما می‌توانید تمام دستورات systemctl زیر را با استفاده از سوئیچ `<host>@<user>-H` برای کنترل systemd به عنوان یک ماشین از راه دور استفاده کنید. این نوع فرمان زمانی که از طریق SSH به یک سیستم از راه دور متصل شده‌اید به کمک شما خواهد آمد.

نکته: systemadm یک برنامه گرافیکی رسمی برای کنترل systemctl است. بستهٔ `systemd-ui-git` را از AUR نصب کنید.

## بررسی وضعیت سیستم

برای دیدن لیستی از واحدهای در حال اجرا فرمان زیر را صادر کنید:

```
$ systemctl
```

یا

```
$ systemctl list-units
```

لیست واحدهایی که با مشکل مواجه شده‌اند:

```
$ systemctl --failed
```

فایل‌های واحدهای در دسترس می‌توانند از مسیر `/usr/lib/systemd/system/` و



`/etc/systemd/system/` مشاهده شود. شما می‌توانید لیستی از فایل‌های واحد نصب شده را با فرمان زیر مشاهده کنید:

```
$ systemctl list-unit-files
```

## استفاده از واحدها

واحدها می‌توانند، سرویس‌ها (`.service`)، نقاط اتصال (`.mount`)، دیوایس‌ها (`.device`) یا سوکت‌ها (`.socket`) باشند. زمانی که از `systemctl` استفاده می‌کنید معمولاً بایستی نام کامل فایل واحد (به همراه پسوند) را مشخص کنید. به عنوان مثال `sshd.socket`. با این حال در برخی موارد شما می‌توانید از نمونه کوتاه شده برای مشخص کردن یک واحد استفاده کنید؛ به موارد زیر دقت کنید:

- اگر شما پسوند واحد را مشخص نکنید `systemctl` به صورت خودکار `.service` را در نظر خواهد گرفت. به عنوان نمونه `netcfg.service` و `netcfg` با هم برابرند.
- نقاط اتصال به صورت خودکار به واحد مورد نظر خود ترجمه می‌شوند. به عنوان مثال `/home` و `home.mount` با هم برابرند.
- مشابه نقاط اتصال، دیوایس‌ها نیز به صورت خودکار به واحد `.device` ترجمه می‌شوند. به عنوان مثال `/dev/sda2` با `dev-sda2.device` برابر است. برای اطلاعات بیشتر `man systemd.unit` را نگاهی بیندازید.

## فرامین

### استفاده از واحدها

فعال کردن یک واحد به صورت درجا:

```
# systemctl start <unit>
```

غیرفعال کردن درجای یک واحد:

```
# systemctl stop <unit>
```

شروع مجدد یک واحد:

```
# systemctl restart <unit>
```

درخواست از یک واحد برای بارگیری مجدد پیکربندی خود:

```
# systemctl reload <unit>
```

نمایش وضعیت یک واحد شامل اطلاعات این که در حال اجرا هست یا خیر:

```
$ systemctl status <unit>
```

بررسی فعال بودن یا نبودن یک واحد:

```
$ systemctl is-enabled <unit>
```

فعال کردن یک واحد برای شروع شدن خودکار در زمان راه اندازی:

```
# systemctl enable <unit>
```

**نکته:** اگر یک سرویس قسمت نصب نداشته باشد این بدان معناست که به صورت خودکار توسط بقیه سرویس‌ها فراخوانی خواهد شد. اما اگر شما نیاز به نصب دستی آن‌ها دارید دستور زیر به شما کمک خواهد کرد؛ جای foo را با نام سرویس عوض کنید:

```
# ln -s /usr/lib/systemd/system/foo.service /etc/systemd/system/graphical.target.wants/
```

غیرفعال کردن یک واحد برای عدم شروع شدن خودکار در زمان راه اندازی:

```
# systemctl disable <unit>
```

نمایش صفحه راهنمای مربوط به واحد (این قابلیت با فایل خود واحد پشتیبانی می‌شود):

```
$ systemctl help <unit>
```

بارگیری مجدد systemd و بررسی واحدهای جدید یا تغییر کرده:

```
# systemctl daemon-reload
```

## مدیریت قدرت

برای مدیریت قدرت وجود polkit لازم است. چنانچه شما در نشست کاربری systemd-logind در یک سیستم محلی آید و نشست دیگری فعال نباشد دستور زیر بدون مجوز ریشه نیز عمل خواهد کرد. در غیر این صورت سیستم بعد از صدور فرمان کلمه عبور ریشه را درخواست می‌کند.

خاموش و شروع مجدد کردن سیستم:

```
$ systemctl reboot
```

خاموش کردن کامل سیستم:

```
$ systemctl poweroff
```

تعليق کردن سیستم<sup>۱</sup>:

```
$ systemctl suspend
```

بردن سیستم به حالت خواب زمستانی<sup>۲</sup>:

```
$ systemctl hibernate
```

بردن سیستم به وضعیت hybrid-sleep:

```
$ systemctl hybrid-sleep
```

## دیمون (DAEMON)

در آرچ لینوکس دیمون‌ها توسط systemd مدیریت می‌شوند. دستور systemctl رابط کاربری برای مدیریت آن‌ها به حساب می‌آید و اطلاعات فایل‌های <service>.service را برای چگونگی اجرای دیمون بازخوانی می‌کند. فایل‌های سرویس در /etc,usr/lib,run/systemd/system نگهداری می‌شود. پیش از این اشاره کاملی به مدیریت کامل واحدها کردیم.

- ۱ Suspend به حالتی گفته می‌شود که اطلاعات بر روی رم آماده است ولی بقیه اجزای سیستم خاموش می‌شود. در این حالت سیستم آماده به کار خواهد بود و به محض روشن کردن آن به محیط سیستم‌عامل باز خواهید گشت.
- ۲ Hibernate در این حالت اطلاعات موجود بر روی رم به حافظه جانبی منتقل شده و سیستم کامل خاموش می‌شود. بعد از روشن کردن اطلاعات به رم انتقال داده شده و سیستم به همان صورت که به خواب زمستانی رفته بود آماده استفاده می‌شود. به یاد داشته باشید برای این کار شما به فضای swap احتیاج خواهید داشت.

فعال کردن شروع سرویس در حین راه‌اندازی:

```
# systemctl enable <service_name>
```

غیرفعال کردن شروع سرویس در حین راه‌اندازی:

```
# systemctl disable <service_name>
```

شروع دستی یک سرویس:

```
# systemctl start <service_name>
```

خاتمه دستی یک سرویس:

```
# systemctl stop <service_name>
```

شروع مجدد یک سرویس:

```
# systemctl restart <service_name>
```

وضعیت یک سرویس:

```
# systemctl status <service_name>
```

لیست کامل دیمون‌ها را می‌توانید از صفحهٔ مربوط به آن ببینید!

## صدا

### ALSA

ALSA جزئی از هسته است و نیازی به نصب آن ندارید. Udev در هنگام راه‌اندازی سیستم سخت‌افزار شما را بررسی و ماژول‌های لازم برای کارت صدا را بارگذاری می‌کند. به هر حال کارت صدای شما باید به خوبی کار کند اما نیاز است آن را از حالت خفه خارج کنید چرا که همهٔ کانال‌ها به صورت پیش‌فرض خفه شده‌اند.

کاربری که می‌خواهد از ALSA استفاده کند بایستی جزء گروه صدا باشد، چنانچه کاربر مورد نظر

شما جزء این گروه نیست دستور زیر را صادر کنید:

```
# gpasswd -a yourusername audio
```

تذکر: تا زمانی که کاربر لاگبک (یک مرتبه خروج و ورود) نکند تغییری مشاهده نخواهد شد.

بسته `alsa-utils` شامل `alsamixer` می‌شود که به شما اجازهٔ پیکربندی دیوایس صدا را از طریق ترمینال می‌دهد. با دستور زیر آن را نصب کنید:

```
# pacman -S alsa-utils
```

اگر شما ابزار `OSS` را برای کار کردن با `dmix` می‌خواهید بایستی بستهٔ `alsa-oss` را نصب کنید:

```
# pacman -S alsa-oss
```

تذکر: از نسخهٔ `udev>=171` ماژول‌های `OSS` به صورت خودکار بارگزاری نمی‌شوند (`snd_seq_oss`, `snd_pcm_oss`, `snd_mixer_oss`).

برای داشتن ویژگی‌های پیشرفتهٔ دیگر نظیر `upmixing/downmixing` ، `high quality resampling` و ... بستهٔ زیر را نصب کنید:

```
# pacman -S alsa-plugins
```

اکنون نوبت به بازکردن کانال‌ها رسیده است؛ آسان‌ترین راه برای بازکردن کانال‌ها استفاده از `alsamixer` است. دستور زیر را در ترمینال با حساب کاربری خود صادر کنید (برای رفتن به حساب کاربری خود از دستور `su - username` استفاده کنید):

```
$ alsamixer
```

برچسب `MM` در زیر هر کانالی به معنی خفه بودن و `OO` به معنی باز بودن آن کانال است. با کلیدهای چپ و راست بین کانال‌ها جابجا شوید و با کلید `M` آن‌ها را از حالت خفه خارج کنید سپس با کلید بالا به حجم صدا اضافه کنید.

برای بررسی صدا از دستور زیر استفاده کنید:

```
$ speaker-test -c 2
```

عدد جلوی c- را بنابر اسپیکر (بلندگوی) خود تغییر دهید. به عنوان مثال برای اسپیکرهای 7.1 دستور زیر را استفاده کنید:

```
$ speaker-test -c 8
```

اگر شما صدایی را دریافت نمی‌کنید مراحل قبل را دو مرتبه مرور کنید و مطمئن شوید همه آن‌ها را به درستی انجام داده‌اید. همچنین مطمئن شوید PCM و MASTER را از حالت خفه خارج کرده‌اید. چنانچه مشکل مرتفع نشد دستور زیر را با مجوز ریشه صادر کنید (برای رفتن به حساب ریشه از -su استفاده کنید):

```
# alsacnf
```

تمام ماژول‌های لازم بایستی به خوبی شناخته و لود شده باشند و اگر از کارت‌های ISA استفاده نکنید نیاز به پیکربندی خاصی نخواهید داشت.

**اخطار:** چنانچه شما یک کارت PCI یا ISAPNP دارید با توجه به مقادیری که alsacnf به فایل modprobe.conf اضافه می‌کند ممکن است udev در تشخیص خودکار دچار مشکل شود.

## OSS

چنانچه ALSA نیازهای شما را برآورده نمی‌کند یا با آن مشکل دارید می‌توانید از OSS استفاده کنید. برای نصب OSS دستور زیر را صادر کنید:

```
# pacman -S oss
```

با دستور فوق OSS نصب شده و اسکریپت نصب اجرا می‌شود (ماژول‌های ALSA موقتاً غیرفعال شده) و ماژول‌های هسته OSS نصب می‌شود. با توجه به این که ALSA در اسکریپت راه‌اندازی به صورت پیش‌فرض فعال است شما بایستی آن را غیرفعال کنید تا با OSS برخورد نداشته باشد. برای این کار بایستی ماژول مربوطه را بلک‌لیست کنید:

```
# nano /etc/modprobe.d/alsa_blacklist.conf
```

```
install soundcore /bin/false
```

بعد از این کار می‌توانید دیمن OSS را برای شروع در راه‌اندازی فعال کنید:

```
# systemctl enable oss.service
```

در اینجا اگر کاربر مورد نظر جزو گروه صوتی نمی‌باشد با دستور زیر آن را به این گروه اضافه کنید:

```
# gpasswd -a username audio
```

اگر OSS توانایی شناسایی کارت شما را نداشت دستورات زیر را صادر کنید:

```
# ossdetect -v
# soundoff && soundon
```

برای امتحان خروجی صدا دقت کنید که حجم صدا بسیار بلند است و ممکن است به گوش شما صدمه بزند بنابراین چنانچه از هدفن استفاده می‌کنید درجه صدا را کم کنید:

```
$ osstest
```

برای کنترل صدا دو برنامه در اختیار شما قرار دارید:

- `ossmix` یک میکسر تحت خط فرمان.
  - `Ossxmix` یک میکسر با رابط گرافیکی.
- برای اطلاعات بیشتر در این زمینه به صفحه OSS مراجعه کنید.

## sudo

sudo را نصب کنید:

```
# pacman -S sudo
```

برای اضافه کردن کاربر به عنوان کاربر `sudo` (یک `sudoer`) دستور `visudo` باید به عنوان ریشه اجرا شود.

به صورت پیش‌فرض دستور `visudo` از ویرایشگر `vi` استفاده می‌کند. اگر شما با این ویرایشگر آشنایی ندارید می‌توانید از متغیر محیطی `EDITOR` برای تغییر ویرایشگر استفاده کنید. به عنوان مثال با دستور زیر `visudo` از `nano` به عنوان ویرایشگر استفاده خواهد کرد:

```
# EDITOR=nano visudo
```

**تذکر:** لطفاً توجه کنید شما در حال تنظیم متغیر و اجرای visudo به صورت همزمان در یک خط هستید و دستور فوق دو فرمان مجزا نیست.

اگر شما با vi مشکلی ندارید و با آن احساس راحتی می‌کنید دلیلی برای استفاده از متغیر EDITOR=nano ندارید:

```
# visudo
```

این دستور فایل `/etc/sudoers` را در یک جلسه کاری ویژه باز می‌کند. visudo یک رونوشت از فایل مذکور را برای ویرایش به صورت موقتی تهیه می‌کند سپس با استفاده از ویرایشگر مربوطه آن را ویرایش کرده و در نهایت آن را بررسی می‌کند. اگر بررسی با موفقیت گذرانده شد فایل موقت بر روی فایل اصلی با مجوزهای صحیح نوشته می‌شود.

**اخطار:** فایل `/etc/sudoers` را با ویرایشگر به صورت مستقیم مورد ویرایش قرار ندهید؛ چرا که در این صورت خطاهای آزاردهنده‌ای انتظار شما را می‌کشد (مانند بلا استفاده شدن حساب کاربری ریشه). شما باید و باید از فرمان visudo برای ویرایش `/etc/sudoer` استفاده کنید.

در قسمت قبلی کاربر ساخته شده را به گروه wheel اضافه کردیم. برای دادن دسترسی‌های ریشه به کاربران این گروه زمانی که از sudo استفاده می‌کنند خط زیر را از حالت کامنت خارج کنید:

```
%wheel ALL=(ALL) ALL
```

اکنون شما می‌توانید به هر کاربری اجازه استفاده از sudo را با اضافه کردن آن کاربر به گروه wheel بدهید. برای اطلاعات بیشتر sudo را ملاحظه نمائید<sup>۱</sup>.

## Yaourt

شاید شما نیاز به نصب برنامه‌هایی از مخزن کاربران آرچ یا همان AUR داشته باشید. Yaourt ابزاریست که امکان نصب بسته‌ها را از مخزن کاربران آرچ بسیار ساده می‌کند. برای نصب آن ابتدا مخزن فرانسه را به انتهای فایل `pacman.conf` اضافه کنید:

برای سیستم ۳۲-بیت:



```
# nano /etc/pacman.conf
```

```
[archlinuxfr]
Server = http://repo.archlinux.fr/i686
```

برای سیستم ۶۴-بیت:

```
# nano /etc/pacman.conf
```

```
[archlinuxfr]
Server = http://repo.archlinux.fr/x86_64
```

اکنون دستور زیر را برای نصب Yaourt صادر کنید:

```
# pacman -Sy yaourt
```

برای نصب برنامه از مخزن کاربران آرچ کافیسست yaourt را احضار کنید، به عنوان مثال برای نصب برنامه ساغر کافیسست دستور زیر را با دسترسی کاربر غیر-ریشه صادر کنید:

```
$ yaourt saaghar
```

در ادامه yaourt گزینه‌های موجود در مخزن کاربران آرچ را برای شما به نمایش می‌گذارد که شما با زدن شماره سمت چپ هر بسته می‌توانید آن را نصب کنید (برای چند بسته بین شماره‌ها فاصله بگذارید).

**تذکر:** با توجه به این که بسیاری از بسته‌ها از طریق کدمنبع واقع در سایت sourceforge ساخته می‌شوند و با توجه به سیاست این سایت مبنی بر تحریم ایران در پاره‌ای از مواقع لازم است که با استفاده از تنظیم پروکسی در ترمینال خود اقدام به نصب از طریق مخزن کاربران آرچ نمائید:

```
$ export http_proxy=http://"proxy_ip":"port_number"
```

به عنوان مثال برای تنظیم پروکسی برنامه your-freedom در ترمینال قبل از به کارگیری yaourt این دستور را صادر کنید و مطمئن شوید your-freedom به خوبی به سرور مورد نظر متصل شده است:

```
$ export http_proxy="http://localhost:8080"
```

برای خارج شدن از این حالت هم دستور زیر را صادر کنید:

```
$ unset http_proxy
```

تذکر: چنانچه تازه‌وارد هستید گام‌های مربوط به ویرایش (Edit) در حین نصب برنامه توسط yaourt را با زدن دکمه n پشت سر بگذارید و مراحل مربوط به ساخت (build) و نصب (install) بسته‌ها را با زدن کلید y تأیید کنید.

## نصب X

X Window System (X11, X) یک پرتوکل نمایش و شبکه سازيست که ایجاد پنجره بر روی سیستم بیت‌مپ را ممکن می‌کند. همچنین پرتوکل و تول‌کیت استاندارد برای ساخت رابط کاربری را نیز فراهم می‌کند (GUIs).

اکنون ما بایستی بسته‌های بیس Xorg را نصب کنیم. حالا نوبت یک‌من است، دستور زیر را به منظور نصب بسته‌های پایه‌ای صادر کنید:

```
# pacman -S xorg-server xorg-xinit xorg-server-utils
```

برای پشتیبانی از 3D بایستی mesa را نصب نمایید:

```
# pacman -S mesa
```

## نصب راه‌انداز ویدویی

نکته: اگر شما آرچ را به صورت مهمان بر روی وی‌باکس نصب کرده‌اید برای کامل کردن نصب X بایستی به راه دیگری متصل شوید. قسمت مربوط به آرچ‌لینوکس مهمان وی‌باکس را ببینید سپس به قسمت پیکربندی در پایین بروید.<sup>۱</sup>

اگر نمی‌دانید که کامپیوتر شما از چه چیپست ویدیویی استفاده می‌کند دستور زیر را اجرا کنید:

```
$ lspci | grep VGA
```

برای دریافت یک لیست کامل از راه‌اندازهای ویدیویی دستور زیر را صادر کنید:

```
$ pacman -Ss xf86-video | less
```

راه‌انداز vesas یک راه‌انداز با عمومی است که تقریباً با همه GPUها سازگار است ولی ممکن است تمام قابلیت‌های دو و سه بعدی را پشتیبانی نکند. اگر بقیه راه‌اندازها با مشکل مواجه شدند یا چیز مناسبی پیدا نکردید vesas می‌تواند گزینه مناسبی باشد:

```
# pacman -S xf86-video-vesa
```

در نهایت برای استفاده از حداکثر قابلیت کارت ویدیویی خود بهتر است بهترین و سازگارترین راه‌انداز را نصب نمائید. در زیر لیستی به این منظور تهیه شده است:

برند	نوع	راه‌انداز	بسته Multilib (برای برنامه‌های ۳۲ بیت بر روی آرچ ۶۴ بیت)
AMD/ATI	منبع باز	xf86-video-ati	lib32-ati-dri
	انحصاری/اختصاصی	catalyst-dkms	lib32-catalyst-utils
Intel	منبع باز	xf86-video-intel	lib32-intel-dri
		xf86-video-i740	
Nvidia	منبع باز	xf86-video-nouveau	lib32-nouveau-dri
		xf86-video-nv	
SiS	منبع باز	nvidia	lib32-nvidia-utils
		xf86-video-sis xf86-video-sisimedia xf86-video-sisusb	

## ATI

دارندگان کارت‌ها گرافیکی ATI دو انتخاب پیش رو خواهند داشت. راه‌انداز اختصاصی کاتالیست

یا راه‌انداز منبع‌باز (xf86-video-ati). راه‌انداز منبع‌باز در مورد کارت‌های ویدیویی جدید و پشتیبانی سه‌بعدی در حد راه‌انداز اختصاصی نیست. با این حال اگر شما واقعاً نمی‌دانید که چه باید بکنید بهتر است ابتدا راه‌انداز منبع‌باز را امتحان کنید چرا که بیشتر نیازهای شما را برطرف خواهد نمود و در دسر کمتری به همراه دارد.

### xf86-video-ati

این راه‌انداز با چیپست‌های HD 6xxx و 7xxxM به بالا به خوبی کار می‌کند.

- سری‌های X1xxx به طور کامل پشتیبانی می‌شوند و شتاب‌دهنده‌های دو-بعدی و سه-بعدی مهیا است.
- شتاب‌دهنده دو-بعدی در سری‌های 2xxx تا HD 6xxx به طور کامل و شتاب‌دهنده سه-بعدی به صورت کاربردی در این سری مهیا است. اما راه‌انداز اختصاصی در این زمینه بهتر عمل خواهد کرد.

به هر حال شما از هر کارت ویدیویی ATI که استفاده می‌کنید xf86-video-ati بایستی انتخاب اول شما باشد. اما در صورتی که شما به راه‌انداز جدیدتر کارت‌های ATI نیازمندید راه‌انداز کاتالیست (اختصاصی) می‌تواند راه‌گشا باشد.

**نکته:** xf86-video-ati با نام radeon برای هسته و xorg.conf شناسایی می‌شود.

### نصب xf86-video-ati

اگر Catalyst/fglrx را قبلاً نصب نموده‌اید بایستی آن‌ها را ابتدا حذف نمائید. سپس اقدام به نصب xf86-video-ati نمائید:

```
# pacman -S xf86-video-ati
```

همچنین یک نسخه git به همراه بقیه بسته‌های مورد نیاز نیز در مخزن کاربران آرچ یا مخزن رادان موجود می‌باشد.<sup>۱</sup>

### بیکربندی

Xorg به صورت خودکار راه‌انداز را بارگیری می‌کند و از EDID مانیتور برای تنظیم تفکیک‌پذیری

صفحه استفاده خواهد کرد. تنظیمات فقط برای تنظیم بیشتر و عمل کرد بهتر انجام می‌پذیرد. اگر شما طلبه یک پیکربندی دستی هستید فایل `/etc/X11/xorg.conf.d/20-radeon.conf` را بسازید و خطوط زیر را به آن اضافه کنید:

```
# nano /etc/X11/xorg.conf.d/20-radeon.conf
```

```
Section "Device"
    Identifier "Radeon"
    Driver "radeon"
EndSection
```

اکنون در فایل فوق می‌توانید تنظیمات سفارشی‌تر شده را اعمال کنید. برای اطلاعات بیشتر در این مورد به صفحهٔ مربوط به ویکی آن مراجعه کنید<sup>۱</sup>.

## Catalyst

برای نصب این راه‌انداز شما راه‌های مختلفی از جمله نصب آن از مخزن رسمی (که شامل legacy driver نمی‌شود)، نصب از مخزن غیر رسمی کاتالیست (به همراه همهٔ بسته‌های لازم) و نصب از مخزن کاربران آرچ (به همراه همهٔ بسته‌های لازم) را در پیش روی خود دارید.

قبل از این که روش نصب را انتخاب کنید لازم است که راه‌اندازی که نیاز دارید را برگزینید. از کاتالیست AMD ۱۲.۴ توسعهٔ کارت‌های Radeon HD 5xxx و Radeon HD 2xxx, 3xxx و 4xxx را از هم جدا نموده است. برای کارت‌های Radeon HD 2xxx, 3xxx and 4xxx راه‌انداز legacy Catalyst و برای Radeon HD 5xxx کاتالیست معمولی موجود است.

## نصب راه‌انداز

### نصب از مخزن رسمی

راه بسیار ساده‌ایست چراکه نیازی به اضافه کردن مخزن غیررسمی ندارید و این بسته از DKMS برای ساخت خودکار ماژول‌های هسته زمانی که هسته به روز شود استفاده می‌کند. اگر این مخزن بسته‌ای که نیاز دارید را دارد وقت را از دست ندهید و همین را نصب کنید.

**نکته:** به یاد داشته باشید که بستهٔ linux-headers را برای کارکرد درست DKMS نصب کرده

باشید.

**نکته:** اگر پک من از شما در مورد پاک کردن libgl سؤال کرد پاسخ مثبت به او بدهید.

**نکته:** اگر در نظر دارید از DKMS استفاده کنید همانگونه که در بخش فعال کردن سرویس systemd گفته شد سرویس آن را فعال کنید.

اگر شما بر روی سیستم ۶۴-بیتی هستید و نیاز به پشتیبانی ۳۲-بیت OpenGL دارید بسته lib32-catalyst-utils را نصب کنید. توجه داشته باشید بایستی مخزن [multilib] فعال باشد:

```
# pacman -S lib32-catalyst-utils
```

### نصب از مخزن غیررسمی (برای راه انداز اختصاصی این روش پیشنهاد می شود)

اگر بسته مورد نظر شما در مخزن رسمی وجود ندارد می توانید از مخزنی که توسط ViOLLO نگهداری می شود به این منظور استفاده کنید. تمام بسته ها امضا شده و استفاده از آن ها بی خطر می باشد.

ViOLLO سه مخزن جداگانه که شامل راه اندازهای مختلف می شوند دارد.

- [catalyst] که راه انداز معمول است و مورد نیاز HD 5xxx و بالاتر می باشد. این مخزن شامل آخرین نسخه پایدار یا بتای کاتالیست می شود.
- [catalyst-stable] که راه انداز معمول است و مورد نیاز HD 5xxx و بالاتر می باشد. این مخزن شامل آخرین نسخه پایدار کاتالیست می شود.
- [catalyst-hd234k] که راه انداز legacy Catalyst است و برای کارت های Radeon HD 2xxx, 3xxx و 4xxx مورد استفاده قرار می گیرد.

**اخطار:** راه انداز Legacy از Xorg 1.13 پشتیبانی نمی کند و چنانچه مایل به نصب آن هستید بایستی توسط راهنمای مربوطه به Xorg 1.12 بازگردید.

برای فعال کردن هر کدام از این مخازن بایستی فایل /etc/pacman.conf را ویرایش کنید:

```
# nano /etc/pacman.conf
```

برای [catalyst]:

```
# nano /etc/pacman.conf
```

```
[catalyst]
Server = http://catalyst.wirephire.com/repo/catalyst/$arch
```

برای [catalyst-stable]:

```
# nano /etc/pacman.conf
```

```
[catalyst-stable]
Server = http://catalyst.wirephire.com/repo/catalyst/$arch
```

برای [catalyst-hd234k]:

```
# nano /etc/pacman.conf
```

```
[catalyst-hd234k]
Server = http://catalyst.wirephire.com/repo/catalyst-hd234k/$arch
```

همچنین شما بایستی کلید GPG را برای اعتماد پک‌من به مخزن اضافه کنید:

```
# pacman-key --keyserver pgp.mit.edu --recv-keys 0xabed422d653c3094
# pacman-key --lsign-key 0xabed422d653c3094
```

بعد از اضافه کردن کلید دیتابیس پک‌من را به روز کنید و بسته را نصب نمائید:

```
# pacman -Syu
# pacman -S catalyst catalyst-utils
```

**نکته:** اگر پک‌من از شما در مورد پاک‌کردن libgl سؤال کرد پاسخ مثبت به او بدهید.

اگر شما بر روی سیستم ۶۴ بیتی هستید و نیاز به پشتیبانی ۳۲-بیت OpenGL دارید بسته lib32-catalyst-utils را نصب کنید. توجه داشته باشید بایستی مخزن [multilib] فعال باشد:

```
# pacman -S lib32-catalyst-utils
```

همهٔ مخزن‌های معرفی شده شامل بقیهٔ بسته‌های مورد نیاز که بسته Catalyst را جایگزین و ماژول

fglrx را فراهم کنند می‌باشند.

- Catalyst-generator: این بسته قادر به ساخت بستهٔ مازول fglrx که توسط پک‌من قابل خواندن باشد، است. ساخت بستهٔ مازول با این بسته بسیار امن است ولی انجامش به صورت دستی امکان‌پذیر است (خودکار نیست).
- Catalyst-hook: یک hook برای mkinitcpio که به صورت خودکار مازول fglrx را بعد از به‌روزرسانی هسته به روز می‌کند؛ کاری مشابه بستهٔ catalyst-dkms در مخزن [community].

### نصب از AUR

برای نصب کاتالیست از AUR می‌توانید از این روش استفاده کنید. باید توجه داشته باشید این روش از همه پرزحمت‌تر است و با هر به‌روزرسانی هسته شما بایستی کاتالیست را مجدداً نصب نمائید.

**اخطار:** اگر شما بستهٔ کاتالیست را از AUR نصب کنید بایستی هر بار پس از به‌روزرسانی هسته آن را مجدداً بسازید در غیر این صورت X در حین راه‌اندازی با مشکل روبرو خواهد شد.

**اخطار:** راه‌انداز Legacy از Xorg 1.13 پشتیبانی نمی‌کند و چنانچه مایل به نصب آن هستید بایستی توسط راهنمای مربوطه به Xorg 1.12 بازگردید.

تمام بسته‌های مورد نیاز در مخزن کاربران موجود می‌باشد:

- Catalyst
- Catalyst-utils
- Lib32-catalyst-utils
- Catalyst-generator
- Catalyst-hook
- Catalyst-daemon

همچنین در AUR چند بسته شامل Catalyst-total و نسخه‌های آزمایشی که در هیچ مخزن دیگری

یافت نمی‌شود نیز موجود است:

- Catalyst-total-hd234k
- Catalyst-total
- Catalyst-test
- Lib32-catalyst-test



• Catalyst-total-pxp  
 بسته Catalyst-total زندگی را به کام شما آسان خواهد نمود چرا که با نصب آن راه‌انداز به علاوه ابزارهای هسته و ابزارهای هسته ۳۲-بیت را خواهید داشت. همچنین Catalyst-hook هم به صورت خودکار ساخته خواهد شد.  
 Catalyst-total-pxp کاتالیست با پشتیبانی آزمایشی powerXpress را نصب می‌نماید.  
 شما همچنین می‌توانید راه‌انداز را مستقیماً از AMD نصب کنید ولی به خاطر مشکلاتی که ممکن است با آن مواجه شوید از توضیح آن صرف نظر می‌کنیم.

## اینتل

با توجه به این که اینتل راه‌اندازهای منبع‌باز را فراهم و پشتیبانی می‌کند به خوبی توسط خود هسته شناسایی و راه‌اندازی می‌شود.  
 توجه داشته باشید سری‌ها GMA 500 و GMA 3600 توسط راه‌اندازهای منبع‌باز پشتیبانی نمی‌شوند.

## نصب

پیش‌نیاز Xorg خواهد بود. برای نصب بسته xf86-video-intel که در مخازن رسمی نیز موجود است از پک‌من کمک بگیرید. این بسته شتاب‌دهنده دو-بعدی و یک راه‌انداز XvMC برای دیکود کردن ویدیو بر روی GPUهای قدیمی‌تر نیز فراهم می‌کند. Intel-dri نیز به عنوان یک پیش‌نیاز نصب خواهد شد و راه‌انداز DRI شتاب‌دهنده سه-بعدی را نیز فراهم می‌کند.

```
# pacman -S xf86-video-intel
```

شتاب‌دهنده سخت‌افزاری دیکود/انکود ویدیویی نیز در GPUهای جدیدتر از طریق راه‌انداز VA-API امکان‌پذیر است. راه‌انداز VA-API توسط بسته libva-intel-driver تأمین می‌شود (در مخزن رسمی موجود است).

```
# pacman -S libva-intel-driver
```

**نکته:** کاربران سیستم‌های ۶۴ بیت نیاز به نصب lib32-intel-dri برای پشتیبانی از شتاب‌دهنده

سه-بعدی در برنامه‌های ۳۲ بیت خواهند داشت.

اطلاعات تخصصی‌تر را در صفحه‌ی ویکی مربوط به کارت‌های اینتل جستجو کنید<sup>۱</sup>.

## NVIDIA

### Nouveau

Nouveau یک راه‌انداز متن‌باز برای کارت‌های گرافیکی انویدیا است<sup>۲</sup>. قبل از انجام هر کاری بهتر است به ماتریکس ویژگی‌ها<sup>۳</sup> نگاهی بیندازید و ویژگی‌هایی که یک معماری مشخص، پشتیبانی می‌کند را بررسی کنید. همچنین نگاهی به نام‌رمزها<sup>۴</sup> برای مشخص کردن دسته‌ی کارت خود بیاندازید. اگر راه‌انداز اختصاصی را نصب نموده‌اید ابتدا بایستی آن را حذف کنید:

```
# pacman -Rdds nvidia nvidia-utils
# pacman -S --asdeps libgl
```

ترفند: چنانچه می‌خواهید راه‌انداز انویدیا را نگه دارید بایستی یک سری تنظیمات را برای بارگیری Nouveau انجام دهید<sup>۵</sup>.

نکته: اگر بعد از حذف راه‌انداز انویدیا اقدام به نصب Nouveau نمودید به یاد داشته باشید که فایل `/etc/X11/xorg.conf` را پاک‌کنید در غیر این صورت X نمی‌تواند راه‌انداز Nouveau را به خوبی بارگیری کند.

همچنین راه‌انداز DDX را با بسته `xf86-video-nouveau` موجود در مخازن رسمی نصب کنید. این کار `nouveau-dri` را به عنوان پیش‌نیاز نصب خواهد نمود تا راه‌انداز DRI برای شتاب‌دهنده‌ی سه-بعدی فراهم باشد.

همچنین مطمئن شوید که Xorg را به خوبی نصب کرده‌اید.

۱ [https://wiki.archlinux.org/index.php/Intel\\_Graphics](https://wiki.archlinux.org/index.php/Intel_Graphics)

۲ <http://nouveau.freedesktop.org/wiki/FAQ>

۳ <http://nouveau.freedesktop.org/wiki/FeatureMatrix>

۴ <http://nouveau.freedesktop.org/wiki/CodeNames>

۵ [https://wiki.archlinux.org/index.php/Nouveau#Keep\\_NVIDIA\\_driver\\_installed](https://wiki.archlinux.org/index.php/Nouveau#Keep_NVIDIA_driver_installed)

برای پشتیبانی سه-بعدی برنامه‌های ۳۲ بیت در سیستم ۶۴ بیت بسته lib32-nouveau-dri را از مخزن multilib نصب نمائید.

## راه‌انداز اختصاصی انویديا

**اخطار:** راه‌انداز اختصاصی انویديا احتمالاً با سیستم‌های EFI کار نخواهد کرد.

### نصب

در اینجا راهنمای نصب راه‌انداز اختصاصی انویديا را برای بسته اصلی لینوکس مشاهده خواهید کرد. برای نصب راه‌انداز در هسته کاستوم شده به بخش مربوط به ویکی مراجعه کنید<sup>۱</sup>.

**ترفند:** نصب راه‌انداز انویديا از طریق پک‌من به صرفه‌تر از نصب بسته‌ای که توسط سایت انویديا تهیه می‌شود است چرا که به راه‌انداز اجازه می‌دهد با به‌روزرسانی سیستم خود را به روز کند.

۱. به قسمت دانلود راه‌انداز در سایت انویديا مراجعه کنید و با استفاده از مدل گرافیک خود راه‌انداز مناسب خود را پیدا کنید<sup>۲</sup>.

**نکته:** برای آخرین مدل‌های GPU شاید لازم باشد که nvidia-beta را از AUR نصب کنید چرا که راه‌انداز پایدار شاید پاسخگوی ویژگی‌های جدید اضافه شده نباشد.

۲. نصب راه‌انداز مناسب کارت شما:

○ برای کارت‌های جی‌فورس سری ۸ یا جدیدتر [NVC0 و جدیدتر] بسته nvidia را از مخازن رسمی نصب نمائید:

```
# pacman -S nvidia
```

○ برای کارت‌های جی‌فورس سری ۶ یا ۷ [NV40-NVAF] بسته nvidia-304xx را از مخازن رسمی نصب نمائید:

```
# pacman -S nvidia-304xx
```

○ برای کارت‌های جی‌فورس سری ۵ اف‌ایکس [NV30-NV38] بسته nvidia-173xx را از مخزن کاربران آرچ نصب نمائید:

```
$ yaourt -S nvidia-173xx
```

۱ <https://www.archlinux.org/packages/?name=linux>

۲ <http://www.nvidia.com/Download/index.aspx?lang=en-us>

○ برای کارت‌های جی فورس سری ۲/۳/۴ ام ایکس/تی‌آی [NV11 and NV17-NV28] بسته nvidia-96xx را از مخزن کاربران آرچ نصب کنید:

```
$ yaourt -S nvidia-96xx
```

Nvidia-utils به عنوان یک پیش‌نیاز برای بسته nvidia نصب خواهد شد و احتمالاً با بسته libgl برخورد خواهد داشت. در این صورت پک‌من از شما می‌پرسد که بسته libgl را حذف کند و چنانچه پاسخ مثبت را بدهید به شما خطای unsatisfied dependencies را نشان خواهد داد. برای رفع این مشکل دستور زیر را اجرا کنید و مجدداً راه‌انداز را نصب نمایید:

```
# pacman -Rdd libgl
```

بسته nvidia-96xx-utils ریلیس ایکس.اورگ قبلی را می‌خواهد (xorg-server1.12) که با xorg-server در مخازن اصلی برخورد خواهد داشت.

نکته: برای آرچ لینوکس ۶۴ بیت حتماً بسته‌های معادل lib32 را نصب کنید. به عنوان مثال بسته‌های lib32-nvidia-utils یا lib32-nvidia-utils-beta یا lib32-nvidia-304xx-utils

ترفند: راه‌اندازهای لگسی nvidia-96xx و nvidia-173xx می‌توانند از طریق مخزن غیررسمی city نصب شوند:

```
# nano /etc/pacman.conf
```

```
[city]
SigLevel = PackageRequired
Server = http://pkgbuild.com/~bgyorgy/$repo/os/$arch
```

۳. بسته انویديا شامل فایل‌سیست که ماژول nouveau را بلک‌لیست می‌کند بنابراین شروع مجدد سیستم لازم است.

بعد از شروع مجدد سیستم چنانچه نیازی به پیکربندی ایکس داشتید دستور زیر را صادر کنید:

```
# nvidia-xconfig
```

یک پیکربندی کمینه برای انویديا به این صورت می‌باشد:

```
# nano /etc/X11/xorg.conf
```

```

Section "Device"
    Identifier "Device0"
    Driver      "nvidia"
    VendorName "NVIDIA Corporation"
EndSection
    
```

اطلاعات بیشتر در مورد نصب و پیکربندی کارت گرافیک را در مدخل ویکی آرچ ملاحظه کنید<sup>۱</sup>.

## نصب راه‌اندازهای ورودی

Udev قادر به شناسایی سخت‌افزار شما خواهد بود. همچنین راه‌انداز `evdev` (`xf86-input-evdev`) یک راه‌انداز ورودی هات-پلاگین مدرن برای تقریباً تمام دیواس‌هاست در نتیجه نیازی به نصب راه‌اندازهای ورودی در بسیاری از مواقع نیست. در اینجا `evdev` به عنوان پیش‌نیاز `xorg-server` نصب شده است.

کاربران لپ‌تاپ یا کاربران صفحات لمسی نیاز به نصب بسته `xf86-input-synaptics` برای کار کردن صفحه لمسی خواهند داشت:

```
# pacman -S xf86-input-synaptics
```

اطلاعات بیشتر برای تنظیمات بیشتر را در مدخل ویکی آرچ ملاحظه نمائید<sup>۲</sup>.

## پیکربندی ایکس

**اخطار:** توجه داشته باشید که بعد از نصب راه‌اندازهای اختصاصی سیستم را معمولاً باید شروع مجدد کنید.

Xorg قابلیت پیکربندی خودکار و در نتیجه کار کردن بدون نیاز به `xorg.conf` را دارد. به هر حال

<sup>۱</sup> <https://wiki.archlinux.org/index.php/NVIDIA>

<sup>۲</sup> [https://wiki.archlinux.org/index.php/Touchpad\\_Synaptics](https://wiki.archlinux.org/index.php/Touchpad_Synaptics)

چنانچه شما مایلید که X Server را به صورت دستی پیکربندی کنید مدخل ویکی<sup>۳</sup> آن در سایت آرچ لینوکس را مشاهده کنید.

## تست X

نکته: این قسمت کاملاً اختیاری است.

نکته: اگر راه اندازهای ورودی در حین تست کار نمی‌کرد، راه اندازهای مورد نیاز را از گروه xorg-drivers نصب کنید و دوباره امتحان کنید. برای دریافت لیست کاملی از این راه اندازها دستور زیر را صادر کنید:

```
$ pacman -Ss xf86-input | less
```

اگر تصمیم گرفتید که هات-پلاگین را غیر فعال کنید تنها کافیست که دو راه انداز xf86-input-keyboard یا xf86-input-mouse را نصب نمائید.

محیط پیش فرض را نصب کنید:

```
# pacman -S xorg-twm xorg-xclock xterm
```

اگر Xorg را قبل از ساختن یک کاربر غیر ریشه نصب کرده باشید در نتیجه یک فایل xinitrc. در دایرکتوری خانگی شما موجود خواهد بود. ابتدا آن را حذف نمائید:

```
$ rm ~/.xinitrc
```

نکته: ایکس بایستی در همان tty که شما لاگین کرده‌اید برای حفاظت از نشست اجرا شود.

برای اجرای نشست Xorg دستور زیر را اجرا کنید:

```
$ startx
```

چند پنجره کوچک قابل جابجا شدن نمایش داده می‌شود. ماوس شما بایستی کار کند. زمانی که خیالتان از بابت همه چیز آسوده شد با صدور فرمان exit از محیط خارج شوید.

```
$ exit
```

اگر صفحه سیاه شد بهتر است به یک کنسول مجازی دیگر بروید (به عنوان مثالاً Ctrl+Alt+F2) و چشم بسته وارد حساب کاربری ریشه شوید (بنویسید root و Enter را بزنید سپس کلمه عبور را وارد کنید و Enter را بزنید. سپس X server را بکشید:

```
# pkill X
```

اگر عمل نکرد چشم بسته دستور شروع مجدد را صادر کنید:

```
# reboot
```

## حل مشکل

اگر مشکلی پیش آمد به فایل Xorg.0.log نگاهی بیندازید. سپس به خطوطی که با (EE) آغاز شده برای یافتن مشکل توجه کنید. خطوطی که با (WW) آغاز شده‌اند نیز شامل اخطارها می‌شوند که می‌تواند نشانگر مشکلی باشد:

```
$ grep EE /var/log/Xorg.0.log
```

اگر شما هنوز مشکل را ریشه‌یابی نکرده‌اید و نیاز به کمک دارید برنامه wgetpaste را نصب کنید و دستورات زیر را برای به در آوردن لینک به مشکل اجرا کنید:

```
# pacman -S wgetpaste
$ wgetpaste ~/.xinitrc
$ wgetpaste /etc/X11/xorg.conf
$ wgetpaste /var/log/Xorg.0.log
```

بعد از هر بار اجرای دستور wgetpaste فایل مورد اشاره آپلود شده و لینکی که شامل اطلاعات آن می‌شود به شما نشان داده خواهد شد. سه لینکی که از دستورات فوق به دست آورده‌اید به علاوه اطلاعات راه‌انداز و سخت‌افزاری سیستم خود را در انجمن‌های آرچ یا کانال IRC برای گرفتن کمک قرار دهید.

## فونت‌ها

در این مرحله شاید بخواهید که سستی از فونت‌های TrueType را نصب کنید. DejaVu یک ست فونت با کیفیت بالاست:

```
# pacman -S ttf-dejavu
```

پیکربندی فونت‌ها می‌تواند در نحوه نمایش بهتر آن‌ها تاثیرگذار باشد. همچنین شما پارسی زبانان می‌توانید ست فونت زیر که گلچینی از بهترین فونت‌های پارسی، توسط وبسایت "یادداشت‌های بیت نیمسوز"<sup>۱</sup> ارائه شده را استفاده کنید. دستورات زیر را برای نصب axel، دریافت و اجرای اسکریپت مربوطه صادر کنید. (تاریخ انقضای این اسکریپت پایان شهریور ماه سال نود و دو می‌باشد. لطفاً برای اجتناب از هرگونه مشکلی این اسکریپت را بعد از این تاریخ اجرا نکنید و نسخه جدیدتر آرچ‌پوک را دریافت کرده و اسکریپت موجود در آن را اجرا کنید)

```
# pacman -S axel
# axel http://emanlog.com/archbook/fonts/farsifonts.sh
# chmod a+x ./farsifonts.sh
# ./farsifonts.sh axel
```

در ادامه اسکریپت از شما می‌خواهد که فونت‌های مورد نظر خود را انتخاب کنید. شما با نوشتن عدد جلوی گزینه ALL.. می‌توانید تمام آن‌ها را نصب کنید. لازم است چند دقیقه‌ای صبر کنید تا اسکریپت کار خود را به پایان برساند.

## اضافه کردن فونت

برای اضافه کردن فونت کافیست که یک دایرکتوری با نام fonts. در خانه (/home) ایجاد کنید<sup>۲</sup> و فونت‌ها را داخل آن مستقر نمایید. به عنوان مثال می‌توانید با مراجعه به سایت font.ir مجموعه‌ای از فونت‌های دلخواه خود را دانلود و به این مسیر منتقل کنید:

```
$ mkdir ~/.fonts
```

بعد از کپی کردن فونت‌ها لازم است که کش فونت را به‌روزرسانی کنید:

۱ [https://wiki.archlinux.org/index.php/Font\\_Configuration](https://wiki.archlinux.org/index.php/Font_Configuration)

۲ <http://cyberrabbits.net>

۳ به نقطه قبل از اسم font دقت کنید. این پوشه یک پوشه مخفی است.



\$ fc-cache -vf

چنانچه می‌خواهید فونت‌ها برای تمام کاربران در دسترس باشد آن‌ها را در مسیر `/usr/share/fonts/` کپی کنید.

## انتخاب و نصب یک رابط گرافیکی

سیستم X Window یک چارچوب پایه‌ای برای ساخت رابط گرافیکی کاربر (GUI) فراهم می‌کند. توجه داشته باشید انتخاب یک DE یا WM یک انتخاب کاملاً شخصی است. بهترین را نسبت به نیاز خود نصب کنید. ما در این کتاب طریقهٔ نصب چند DE و WM محبوب‌تر را پوشش می‌دهیم.

- مدیر پنجره (WM)<sup>۱</sup>  
مدیر پنجره وظیفه کنترل کردن موقعیت و ظاهر پنجرهٔ ابزارها با اتصال به سیستم X Window را بر عهده دارد. قسمت مدیرهای پنجره را برای اطلاعات بیشتر مطالعه نمائید<sup>۲</sup>.

- محیط دسکتاپ (DE)<sup>۳</sup>  
محیط دسکتاپ یک رابط کارا و پویاست که اکثر کاربران آن را به مدیرهای پنجره ترجیح می‌دهند. یک DE به طور معمول یک مدیر پنجره، آیکن‌ها، اپلت‌ها، پنجره‌ها، نوار ابزارها، پوشه‌ها، پس‌زمینه‌ها، یک دست از ابزارها و قابلیت کشیدن و رها کردن را فراهم می‌کند. برای اطلاعات بیشتر محیط‌های دسکتاپ را ببینید<sup>۴</sup>.

در ادامه طریقهٔ نصب دسکتاپ‌های KDE، Gnome، LXDE و Xfce به همراه مدیر پنجرهٔ OpenBox و WMFS را آموزش خواهیم داد؛ شما می‌توانید دیگر دسکتاپ‌ها یا مدیرهای پنجره را با مراجعه به لینک‌های مربوطه به راحتی نصب کنید.

## مدیر نمایش

برای راه‌اندازی یک رابط گرافیکی شما به صورت پایه‌ای دو راه پیش رو دارید. یا استفاده از مدیر

<sup>۱</sup> [https://wiki.archlinux.org/index.php/Window\\_Manager](https://wiki.archlinux.org/index.php/Window_Manager)

<sup>۲</sup> [https://wiki.archlinux.org/index.php/Window\\_Manager](https://wiki.archlinux.org/index.php/Window_Manager)

<sup>۳</sup> [https://wiki.archlinux.org/index.php/Desktop\\_Environment](https://wiki.archlinux.org/index.php/Desktop_Environment)

<sup>۴</sup> [https://wiki.archlinux.org/index.php/Desktop\\_Environment](https://wiki.archlinux.org/index.php/Desktop_Environment)

نمایش یا استفاده از `xinitrc`. مدیر نمایش یا مدیر ورود، یک رابط گرافیکی است که بعد از اتمام مراحل راه‌اندازی سیستم نمایش داده می‌شود. سپس شما از طریق آن با وارد کردن مشخصات کاربر و همچنین نشست مورد نظر وارد محیط کاربری خود خواهید شد.

در اینجا شما نحوه اضافه کردن انواع مدیرهای نمایش را برای اجرا به صورت خودکار در حین راه‌اندازی سیستم خواهید آموخت.

**ترفند:** بهتر است از مدیر نمایشی که با محیط دسکتاپ شما (به عنوان مثال `kdm` برای محیط دسکتاپ KDE) هماهنگ است استفاده کنید.

**ترفند:** مدیرهای پنجره زیر قادر به شناسایی خودکار مدیرهای پنجره نصب شده و محیط‌های دسکتاپ هستند: `GDM, KDM, LXDM, LightDM`.

- `GDM` مدیر نمایشی که با دسکتاپ گنوم عرضه می‌شود. برای فعال کردن آن دستور زیر را صادر کنید:

```
# systemctl enable gdm.service
```

- `KDM` مدیر نمایشی که با دسکتاپ KDE عرضه می‌شود. برای فعال کردن آن دستور زیر را صادر کنید:

```
# systemctl enable kdm.service
```

- `SLiM` کی مدیر نمایش سبک که محتویات فایل `~/xinitrc` را برای شروع یک محیط دسکتاپ یا مدیر پنجره می‌خواند (به قسمت ویرایش `~/xinitrc` مراجعه کنید). برای فعال کردن آن دستور زیر را صادر کنید:

```
# systemctl enable slim.service
```

- `LXDM` مدیر نمایشی که می‌تواند مستقل از دسکتاپ `LXDE` نیز اجرا شود. برای فعال کردن آن دستور زیر را صادر کنید:

```
# systemctl enable lxdm.service
```

- `LightDM` مدیر نمایش مناسب و بسیار سبک با قابلیت شناسایی خودکار محیط‌های دسکتاپ و مدیرهای پنجره. برای فعال کردن آن دستور زیر را صادر کنید:

```
# systemctl enable lightdm.service
```

## ویرایش ~/.xinitrc

شما با ویرایش این فایل می‌توانید مدیر پنجره یا محیط دسکتاپ خود را توسط دستور startx اجرا کنید. فایل زیر را باز کنید و محتویات آن را اینگونه تغییر دهید:

```
$ nano ~/.xinitrc
```

```
#!/bin/sh

#
# ~/.xinitrc
#
# Executed by startx (run your window manager from here)
#

exec <session-command>
```

در فایل فوق عبارت <session-command> را با توجه به نیاز خود تغییر دهید. به عنوان مثال برای شروع به کار مدیر پنجره awesome:

```
$ nano ~/.xinitrc
```

```
#!/bin/sh

#
# ~/.xinitrc
#
# Executed by startx (run your window manager from here)
#

exec awesome
```

در ادامه لیست کاملی از دستورات نشست مورد نظر را مشاهده می‌کنید:

```
exec awesome
exec dwm
exec startfluxbox
exec fvwm2
exec gnome-session
exec openbox-session
exec startkde
exec startlxde
exec startxfce4
exec enlightenment_start
```

```
exec mate-session
exec wmfs
```

## نصب و راه اندازی KDE

نکات مهم قبل از نصب:

- مطمئن شوید آیینیهایی که استفاده می‌کنید به‌روزند.
- برای به‌روزرسانی از سوئیچ f جهت فشار آوردن به پک‌من در صورتی که بسته‌ها برخورد داشتند استفاده نکنید و به عنوان باگ‌آن را گزارش دهید.

### نصب KDE

#### نصب کامل

برای نصب KDE ابتدا مطمئن شوید که سیستم شما کاملاً به‌روز است:

```
# pacman -Syu
```

اکنون نوبت به نصب خود KDE است:

```
# pacman -S kde
```

و یا kde-meta:

```
# pacman -S kde-meta
```

تفاوت‌های KDE با KDE-meta را از در مدخل ویکی آن<sup>۱</sup> بخوانید؛ پیشنهاد ما نصب KDE است. اگر شما به فایل‌های زبانی نیاز دارید این بسته را نیز نصب کنید:

```
# pacman -S kde-l10n-yourlanguagehere
```

به عنوان مثال kde-l10n-fa برای زبان پارسی است. همچنین می‌توانید با دستور زیر تمام زبان‌های موجود را ملاحظه کنید:

```
$ pacman -Ss kde-l10n
```

**تذکر:** KDE 4.x ماژولار است؛ شما می‌توانید فقط ابزارهای مورد نظر خود را بدون نیاز به نصب ست کامل بسته‌های KDE نصب کنید.

**تذکر:** KDE 4.x نیازی به بسته‌های Gamin یا Fam برای بررسی تغییرات بر روی فایل‌ها و مسیرهای محلی را ندارد.

### نصب کمینه

اگر شما یک نصب کمینه از KDE SC را می‌خواهید ما نمونه زیر را پیشنهاد می‌کنیم:

```
# pacman -S kdebase phonon-vlc
```

شما می‌توانید از phonon-gstreamer به جای phonon-vlc استفاده کنید.

**تذکر:** به یاد داشته باشید که شما نیاز به نصب یک بسته \*ttf دارید. phonon-vlc به صورت بالقوه احتیاج به ttf-freefonts دارد ولی چنانچه از phonon-gstreamer استفاده می‌کنید بایستی ttf-dejavu یا یک بسته \*ttf دیگر را خودتان نصب کنید.

### راه‌اندازی KDE

شما دو راه اصولی برای اجرای KDE در اختیار دارید. استفاده از مدیر نمایش و یا xinitrc. برای استفاده از مدیر نمایش kdm همانگونه که پیش‌تر اشاره شد بعد از نصب KDE سرویس آن را توسط systemctl فعال کنید. بعد از شروع مجدد سیستم شما به صفحه مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر Slim یا LightDM استفاده کنید. برای راه‌اندازی KDE توسط دستور startx به قسمت ویرایش ~/.xinitrc مراجعه کنید. اطلاعات بیشتر در مورد شخصی‌سازی KDE را در مدخل ویکی<sup>۱</sup> یا پست یک چشمه از KDE<sup>۲</sup> مشاهده نمایید.

### نصب و راه‌اندازی گنوم

گنوم ۳ دو رابط دارد: گنوم شل یک طرح استاندارد و fallback mode به عنوان رابط دوم.

۱ <https://wiki.archlinux.org/index.php/Kde>

۲ <http://linuxreview.ir/1390/05/an-example-of-kdes-flexibility-and-power>

Gnome-session به صورت خودکار کامپیوتر شما را بررسی می‌کند و در صورتی که شرایط را برای گنوم شل مساعد نبیند fallback mode را اجرا می‌کند. fallback mode چیزی شبیه به گنوم ۲ است که از gnome-panel/Metacity به جای gnome-shell/Mutter استفاده می‌کند.

## نصب گنوم

ابتدا سیستم خود را کاملاً به‌روزرسانی کنید:

```
# pacman -Syu
```

سپس با دستور زیر نصب گنوم به علاوهٔ اضافات را آغاز کنید:

```
# pacman -S gnome gnome-extra
```

## راه‌اندازی گنوم

شما دو راه اصولی برای اجرای گنوم در اختیار دارید. استفاده از مدیر نمایش و یا xinitrc. برای استفاده از مدیر نمایش gdm همانگونه که پیش‌تر اشاره شد بعد از نصب گنوم سرویس آن را توسط systemctl فعال کنید. بعد از شروع مجدد سیستم شما به صفحهٔ مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر Slim یا LightDM استفاده کنید. برای راه‌اندازی گنوم توسط دستور startx به قسمت ویرایش ~/.xinitrc مراجعه کنید.

برگهٔ تقلب گنوم شل برای آشنایی بیشتر با این DE می‌تواند بسیار مفید باشد<sup>۱</sup>.

در پاره‌ای از مواقع ممکن است از شما خواسته شود و یا شما نیاز داشته باشید که گنوم شل را مجدداً راه‌اندازی کنید، برای این‌کار کافیست کلیدهای Alt+F2 را فشار داده سپس به ترتیب کلیدهای r و Enter را بر روی صفحه‌کلید فشار دهید.

## نصب و راه‌اندازی MATE

MATE یک فورک از گنوم ۲ است که در حال حاضر بر روی گیت‌هاب توسعه داده می‌شود. بستهٔ پایدار، با شمارهٔ نسخهٔ انتشار در <http://repo.mate-desktop.org/archlinux> میزبانی می‌شود.

## نصب

برای نصب نسخه پایدار میّت توسط پک‌من، خطوط زیر را به `/etc/pacman.conf` اضافه کنید:

```
# nano /etc/pacman.conf

[mate]
Server = http://repo.mate-desktop.org/archlinux/$arch
```

سپس دستور زیر را اجرا کنید:

```
# pacman -Syy
# pacman -S mate
# pacman -S mate-extras
```

در هنگام نصب، احتمالاً شما با یک سری تداخلات مواجه می‌شوید که برای حل آن‌ها باید فایل‌های مورد نظر را تغییر نام دهید و یا با احتیاط از پرچم `-force` استفاده کنید.

## راه‌اندازی MATE

شما دو راه اصولی برای اجرای میّت در اختیار دارید. استفاده از مدیر نمایش و یا `.xinitrc`. برای استفاده از مدیر نمایش `lightdm` ابتدا آن را نصب کنید سپس همانگونه که پیش‌تر اشاره شد بعد از نصب میّت سرویس آن را توسط `systemctl` فعال کنید. بعد از شروع مجدد سیستم شما به صفحه مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر `Slim` استفاده کنید. برای فعال کردن تعویض کاربر در `lightdm` فرمان زیر را اجرا کنید:

```
# ln -s /usr/lib/lightdm/lightdm/gdmflexiserver /usr/bin/mdmflexiserver
```

برای راه‌اندازی میّت توسط دستور `startx` به قسمت ویرایش `~/.xinitrc` مراجعه کنید. لازم است بدانید که بسیاری از برنامه‌های هسته گنوم با پیچیده شدن مجدد برای میّت تغییر نام داده‌اند:

- Nautilus تغییر نام داده شده به Caja
- Metacity تغییر نام داده شده به Marco
- Gconf تغییر نام داده شده به Mate-conf
- Gedit تغییر نام داده شده به Pluma

- Eye of GNOME تغییر نام داده شده به Eye of MATE
- Evince تغییر نام داده شده به Atril
- File Roller تغییر نام داده شده به Engrampa
- GNOME Terminal تغییر نام داده شده به MATE Terminal

## نصب و راه‌اندازی LXDE

### نصب LXDE

LXDE بسیار ماژولار است بنابراین بسته‌هایی که نیاز دارید را فقط نصب خواهید کرد. بسته‌های پایه‌ای برای نصب LXDE عبارتند از lxde-common, lxsession, desktop-file-utils و یک مدیر پنجره. شما می‌توانید گروه LXDE را با دستور زیر نصب کنید:

```
# pacman -Syu lxde
```

این کار بسته‌های زیر را برای شما نصب می‌کند.

Gpicview, libfm, lxappearance, lxappearance-obconf, lxde-common, lxde-icon-theme, lxdm, lxinput, lxlauncher, lxmenu-data, lxmusic, lxpanel, lxrandr, lxsession, lxshortcut, lxtask, lxterminal, menu-cache, openbox, pcmanfm,

menu.xml, rc.xml و autostart را در ~/.config/openbox قرار دهید. این فایل‌ها در مسیر

/etc/xdg/openbox در دسترسند:

```
$ mkdir -p ~/.config/openbox
$ cp /etc/xdg/openbox/{rc.xml,menu.xml,autostart} ~/.config/openbox
```

شما همچنین به نصب Gamin به عنوان یک ابزار مانیتورینگ فایل و مسیر نیاز دارید. این ابزار زمانی که برنامه‌ای به آن نیاز داشته باشد اجرا می‌شود و مانند FAM نیاز به daemon ندارد:

```
# pacman -S gamin
```

شما همچنین شاید نیاز به یک سری از ابزارهای سبک برای این DE دارید. فرمان زیر را برای

نصب یک سری از ابزارهای خیلی ابتدایی صادر کنید:

```
# pacman -S leafpad xarchiver obconf epdfview
```



برخی از بسته‌های LXDE آزمایشی هستند و شما می‌توانید از مخزن کاربران آرچ اقدام به نصب آن‌ها کنید.

## راه‌اندازی LXDE

شما دو راه اصولی برای اجرای LXDE در اختیار دارید. استفاده از مدیر نمایش و یا `xinitrc`. برای استفاده از مدیر نمایش `lightdm` ابتدا آن را نصب کنید سپس همانگونه که پیش‌تر اشاره شد بعد از نصب LXDE سرویس آن را توسط `systemctl` فعال کنید. بعد از شروع مجدد سیستم شما به صفحه مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر `Slim` استفاده کنید. برای راه‌اندازی LXDE توسط دستور `startx` به قسمت ویرایش `~/.xinitrc` مراجعه کنید. اطلاعات بیشتر را در مدخل ویکی مشاهده کنید.<sup>۱</sup>

## نصب و راه‌اندازی Xfce

Xfce یک محیط دسکتاپ دیگر است که منابع کمتری را نسبت به KDE و گنوم مصرف می‌کند. اکثر تنظیمات از طریق رابط گرافیکی انجام می‌پذیرد. `Xfce4` یک نرم‌افزار بالغ و پایدار است. Xfce تا حدودی ماژولار است. به این معنی که شما احتیاج به اجرای تمام قسمت‌ها را در آن واحد نخواهید داشت و می‌توانید از بین آن‌ها انتخاب‌های خاص خود را داشته باشید به همین دلیل Xfce چند بسته جدا برای آرچ دارد.

## نصب Xfce

برای نصب دستور زیر را اجرا کنید:

```
# pacman -S xfce4
```

پس از شما در مورد انتخاب بسته‌هایی که قرار است نصب شود سؤال خواهد کرد؛ چنانچه شما تمامی بسته‌ها را می‌خواهید به راحتی `Enter` را فشار دهید. در غیر این صورت با زدن شماره بسته آن را نصب کنید (بین شماره بسته‌ها را با فاصله مشخص کنید).

به هر حال شما می‌توانید فقط چند بسته پایه‌ای را به جای نصب تمام آن‌ها نصب کنید:

```
# pacman -S xfwm4 xfce4-panel xfdesktop thunar xfce4-session
```

```
# pacman -S xfce4-settings xfce4-appfinder xfce-utils xfconf
```

اگر اضافت را می‌خواهید:

```
# pacman -S xfce4-goodies
```

بسته زیر را هم برای کارکردن xfce4-mixer با ALSA نصب کنید:

```
# pacman -S gstreamer0.10-base-plugins
```

در آخر بسته Gamin را نصب کنید:

```
# pacman -S gamin
```

## راه‌اندازی Xfce

شما دو راه اصولی برای اجرای Xfce در اختیار دارید. استفاده از مدیر نمایش و یا xinitrc. برای استفاده از مدیر نمایش lightdm ابتدا آن را نصب کنید سپس همانگونه که پیش‌تر اشاره شد بعد از نصب Xfce سرویس آن را توسط systemctl فعال کنید. بعد از شروع مجدد سیستم شما به صفحه مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر Slim استفاده کنید. برای راه‌اندازی Xfce توسط دستور startx به قسمت ویرایش ~/.xinitrc مراجعه کنید. برای اطلاعات بیشتر به مدخل ویکی xfce مراجعه کنید<sup>۱</sup>.

## نصب و راه‌اندازی OpenBox

OpenBox یک مدیر پنجره floating بسیار سبک است که از طریق مخزن community در دسترس قرار دارد.

### نصب OpenBox

برای نصب کفایت دستور زیر را صادر کنید:

```
# pacman -S openbox
```

بعد از نصب وارد حساب کاربری خود شوید و دستورات زیر را صادر کنید:

```
$ mkdir -p ~/.config/openbox
$ cp /etc/xdg/openbox/{rc.xml,menu.xml,autostart,environment} ~/.config/openbox
```

## راه‌اندازی OpenBox

شما دو راه اصولی برای اجرای OpenBox در اختیار دارید. استفاده از مدیر نمایش و یا `xinitrc`. برای استفاده از مدیر نمایش `lightdm` ابتدا آن را نصب کنید سپس همانگونه که پیش‌تر اشاره شد بعد از نصب OpenBox سرویس آن را توسط `systemctl` فعال کنید. بعد از شروع مجدد سیستم شما به صفحه مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر `SLiM` استفاده کنید.

برای راه‌اندازی OpenBox توسط دستور `startx` به قسمت ویرایش `~/.xinitrc` مراجعه کنید. برای اطلاعات بیشتر به مدخل ویکی OpenBox<sup>۱</sup> یا آموزش جامع آن به زبان فارسی<sup>۲</sup> مراجعه کنید.

## نصب و راه‌اندازی WMFS

یک مدیر پنجره بی‌نهایت سریع و سبک `Dynamic` است که می‌توانید از آن بر روی سیستم‌های مختلف از رده بسیار ضعیف تا مدرن استفاده کنید.

### نصب WMFS

برای نصب کفایت دستور زیر را صادر کنید، WMFS از طریق مخزن کاربران آرچ در دسترس است:

```
$ yaourt -S wmfs-git
```

بعد از نصب وارد حساب کاربری خود شوید و فایل‌های زیر را به مسیر مورد نظر کپی کنید:

```
$ mkdir -p ~/.config/wmfs
```

<sup>۱</sup> <https://wiki.archlinux.org/index.php/Openbox>

<sup>۲</sup> <https://linuxreview.ir/1390/03/openbox-how-to>

```
$ cp /etc/xdg/wmfs/wmfsrc ~/.config/wmfs
```

## راه‌اندازی WMFS

شما دو راه اصولی برای اجرای WMFS در اختیار دارید. استفاده از مدیر نمایش و یا xinitrc. برای استفاده از مدیر نمایش lightdm ابتدا آن را نصب کنید سپس همانگونه که پیش‌تر اشاره شد بعد از نصب WMFS سرویس آن را توسط systemctl فعال کنید. بعد از شروع مجدد سیستم شما به صفحه مدیر نمایش هدایت خواهید شد. همچنین می‌توانید از مدیرهای نمایش دیگر نظیر Slim استفاده کنید. برای راه‌اندازی WMFS توسط دستور startx به قسمت ویرایش ~/.xinitrc مراجعه کنید. برای اطلاعات بیشتر به مدخل ویکی WMFS مراجعه کنید.

## دانگرید برنامه (اسکرپت دانگرید)

زمانی که یک برنامه را به‌روزرسانی می‌کنید ممکن است با مشکلاتی نیز مواجه شوید که در برخی از مواقع حتی راه‌اندازی سیستم را با اختلال همراه کند. در چنین مواقعی بایستی برنامه مورد نظر را به نسخه قبلی، به اصطلاح دانگرید کنید. به عبارت دیگر نسخه جدید برنامه را حذف و نسخه قدیمی که به خوبی کار می‌کرد را به جای آن نصب کنید. پک‌من به صورت پیش‌فرض بسته‌های دانلود شده را در کش خود نگه می‌دارد و این کار امکان دانگرید آسان برنامه‌ها را برای ما به ارمغان می‌آورد. برای دانگرید چنانچه به سیستم آرچ دسترسی دارید با پک‌من وارد عمل شوید ولی اگر سیستم شما با مشکل جدی روبرو شده بایستی از یک دیسک زنده به سیستم آرچ خود chroot زده و سپس برنامه مورد نظر دانگرید را کنید. پک‌من با استفاده از سوئیچ -l می‌تواند بسته‌های آماده شده برای خودش را نصب کند. پس کفایت به مسیر کش پک‌من بروید و بسته قبلی برنامه را نصب کنید:

```
# cd /var/cache/pacman/pkg/
# pacman -U PKGNAME-VERSION.pkg.tar.xz
```

به عنوان مثال برای دانگرید برنامه icu به بسته icu-4.8-1-i686.pkg.tar.xz کفایت بعد از ورود به پوشه کش، دستور زیر را صادر کنید:

```
# pacman -U icu-4.8-1-i686.pkg.tar.xz
```

همچنین می‌توانید از اسکرپت دانگرید که به این منظور تهیه شده استفاده کنید. برای نصب آن دستور زیر را صادر کنید:

```
$ yaourt -S downgrade
```

بعد از نصب آن کافیسیت برنامه مورد نظر را توسط این اسکرپت به نسخه مورد نظرتان ارتقا دهید یا دانگرید کنید. این اسکرپت ابتدا کَش پک‌من شما را جستجوی می‌کند و از شما می‌خواهد در صورت نیاز با وارد کردن عدد مربوط به بسته دانگرید را انجام دهید در غیر این صورت و چنانچه نسخه مورد نظر را در کَش پیدا نکردید از شما برای جستجوی مخزن A.R.M. سؤال می‌کند که با وارد کردن کلمه s و زدن Enter این کار را برای شما انجام می‌دهد. در ادامه با وارد کردن عدد مربوط به بسته مورد نظر شما عملیات دانلود و نصب بسته آغاز می‌شود. به عنوان مثال برای دانگرید یا ارتقا فایرفاکس کافیسیت دستور زیر را وارد کنید (به خروجی دقت کنید):

```
# downgrade firefox
```

The following packages are available in your cache:

```
1 local firefox-7.0.1-1-i686.pkg.tar.xz
please choose a version, [s]earch A.R.M., or [q]uit: s
```

The following packages are available from the A.R.M.:

```
1 extra firefox-8.0.1-i686.pkg.tar.xz.sig [installed]
2 extra firefox-8.0.1-i686.pkg.tar.xz [installed]
3 extra firefox-7.0.1-i686.pkg.tar.xz.sig
4 extra firefox-7.0.1-i686.pkg.tar.xz
5 extra firefox-7.0.1-1-i686.pkg.tar.xz.sig
6 extra firefox-7.0.1-1-i686.pkg.tar.xz
7 extra firefox-6.0.2-1-i686.pkg.tar.xz.sig
8 extra firefox-6.0.2-1-i686.pkg.tar.xz
9 extra firefox-6.0-1-i686.pkg.tar.xz.sig
10 extra firefox-6.0-1-i686.pkg.tar.xz
11 extra firefox-6.0.1-1-i686.pkg.tar.xz.sig
12 extra firefox-6.0.1-1-i686.pkg.tar.xz
13 extra firefox-5.0-1-i686.pkg.tar.xz
14 extra firefox-4.0-1-i686.pkg.tar.xz
15 extra firefox-4.0.1-1-i686.pkg.tar.xz.sig
16 extra firefox-4.0.1-1-i686.pkg.tar.xz
17 extra firefox-3.6.9-1-i686.pkg.tar.xz
18 extra firefox-3.6.8-1-i686.pkg.tar.xz
19 extra firefox-3.6.7-1-i686.pkg.tar.xz
20 extra firefox-3.6.6-1-i686.pkg.tar.xz
21 extra firefox-3.6.4-1-i686.pkg.tar.xz
22 extra firefox-3.6.3-1-i686.pkg.tar.xz
```

```

23 extra firefox-3.6.2-i686.pkg.tar.gz
24 extra firefox-3.6.2-1-i686.pkg.tar.xz
25 extra firefox-3.6.15-1-i686.pkg.tar.xz
26 extra firefox-3.6.14-1-i686.pkg.tar.xz
27 extra firefox-3.6.13-1-i686.pkg.tar.xz
28 extra firefox-3.6.12-1-i686.pkg.tar.xz
29 extra firefox-3.6.11-1-i686.pkg.tar.xz
30 extra firefox-3.6.10-1-i686.pkg.tar.xz
31 extra firefox-3.5.7-1-i686.pkg.tar.gz
32 extra firefox-3.5.6-1-i686.pkg.tar.gz
33 extra firefox-3.5.5-1-i686.pkg.tar.gz
34 extra firefox-3.5.4-1-i686.pkg.tar.gz
  
```

please choose a version or [q]uit: 2

در مثال فوق با وارد کردن 2 بستهٔ `firefox-8.0-1-i686.pkg.tar.xz` نصب می‌شود. برای پاک‌کردن گش پک‌من از بسته‌هایی که بر روی سیستم شما نصب نیستند می‌توانید از دستور زیر استفاده کنید:

```
# pacman -Sc
```

برای پاک‌کردن کلیه بسته‌های موجود در گش پک‌من از دستور زیر استفاده کنید:

```
# pacman -Scc
```

## کار با chroot

### تغییر ریشه (change root)

Chroot در سیستم عامل یونیکس، عملیاتی است که دایرکتوری ریشه را برای فرآیند در حال اجرا به همراه تمام زیر مجموعه‌های دایرکتوری به دایرکتوری دیگر منتقل می‌کند (تمام فرآیندها و وابستگی‌ها).

**نکته:** وقتی مکان ریشه را به شاخهٔ دیگری تغییر می‌دهید دیگر به دستورات و فایل‌های خارج از آن دایرکتوری دسترسی ندارید. چنین دایرکتوری `chroot jail` یا زندان `chroot` خوانده می‌شود.

Chroot قالباً برای مراقبت از سیستم در مواقعی مثل نصب مجدد گراب (بوت‌لودر) یا ریست کردن پسورد فراموش شده استفاده می‌شود. این عملیات معمولاً از طریق دیسک زنده (Live CD) و یا توسط فلش (Live USB) به یک پارتیشن `mount` شده حاوی سیستم عامل انجام می‌شود.

## پیش نیازها:

وارد شدن به یک گنو/لینوکس (نصب شده یا دیسک زنده) دیگر با معماری مشابه. دسترسی به کاربر ریشه برای انجام عملیات chroot. مطمئن شدن از یکی بودن معماری لینوکس نصب شده با معماری لینوکس بوت شده برای chroot. اگر معماری لینوکس خود را نمی دانید می توانید با زدن دستور زیر در ترمینال از آن آگاه شوید:

```
# uname -m
```

اگر به بارگزاری ماژول خاصی در محیط chroot نیاز دارید، قبل از عملیات chroot آن‌ها را بارگزاری کنید. همچنین خالی کردن swap قبل از عملیات chroot می تواند مفید باشد. برای خالی کردن swap به شیوه زیر عمل کنید:

```
# swapon /dev/<اسم دستگاه یا پارتیشن>
```

## سوار/پیاده کردن سیستم پرونده

در سیستم عامل‌های شبه UNIX همه چیز به صورت یک فایل در یک محل پیش‌بینی شده در سلسله مراتب سیستم ذخیره می‌شود (حتی پوشه‌ها نوع خاصی از فایل‌ها هستند). کلید فضای قابل دسترس برای کاربر در یک درخت دایرکتوری خلاصه می‌شود، پایه این سیستم بر دایرکتوری ریشه استوار است. نصب/سوار کردن (Mounting) یک سیستم پرونده باعث می‌شود تا Linux بداند دستگاه ورودی حاوی چه پرونده و دایرکتوری‌هایی می‌باشد. این پروسه شبیه سوار اسب شدن است با این تفاوت که البته در اینجا اسبی وجود ندارد!

سیستم پرونده را می‌توان سوار/پیاده نمود. به این معنا که سیستم‌های پرونده می‌توانند به درخت دایرکتوری وصل شده و یا اتصال آن‌ها به درخت دایرکتوری را قطع کرد. سیستم‌های پرونده دیگر را نیز در صورت نیاز می‌توان سوار نمود مانند سیستم پرونده‌هایی که درون یک دیوایس می‌باشند.

پروسه سوار کردن یکی از تفاوت‌های بارز بین یونیکس و ویندوز است. برای استفاده از یک دیسک در ویندوز شما باید دیسک را داخل درایو مربوطه قرار دهید، سپس بلافاصله به محتویات آن دسترسی پیدا می‌کنید. ولی در یونیکس بعد از قرار دادن دیسک در درایو باید آن را به طور صحیح سوار نمایید. شاید این کار تا اندازه‌ای پیچیده به نظر برسد ولی در عمل به سهولت انجام می‌گیرد.

برای سوار کردن ابتدا باید مکان ریشه را فهمید، اگر آدرس ریشه را فراموش کرده‌اید دستور زیر را برای فهمیدن آن تایپ کنید (علامت ستاره در خروجی نشانگر پارتیشن ریشه است):

```
# fdisk -l
```

بعد از فهمیدن مسیر پارتیشن باید یک دایرکتوری برای سوار کردن پارتیشن روت ایجاد کرد:

```
# mkdir /mnt/arch
# mount /dev/<اسم دستگاه یا پارتیشن> /mnt/arch
```

## تغییر ریشه

مرحله اول: mount موقت فایل سیستم:

```
# cd /mnt/arch
# mount -t proc proc proc/
# mount -t sysfs sys sys/
# mount -o bind /dev dev/
```

اگر شما به سوار کردن یک شاخه که به صورت خارجی به روت وصل شده (مثل: /usr, /var, /boot و...) نیاز دارید به صورت زیر عمل کنید (به عنوان مثال برای دایرکتوری بوت):

```
# mount /dev/<اسم دستگاه یا پارتیشن> boot/
```

امکان سوار کردن پارتیشن بعد از chroot نیز وجود دارد، ولی در کل این کار پیشنهاد نمی‌شود. دلیل آن این است که بعد خارج شدن از chroot شما باید فایل سیستم‌های mount شده را از سیستم به طور امن جدا کنید (umount) پس اگر قبل از ورود به chroot پارتیشن‌ها را mount کنید، بعد از خروج می‌توانید با دستور (umount all) همه پارتیشن‌ها را از سیستم به طور امن جدا کنید (چون محیط اصلی سیستم همه پارتیشن‌ها را می‌شناسد) ولی اگر بعد از ورود به chroot پارتیشن‌ها را mount کنید امکان umount وجود نخواهد داشت. این کار به ایمنی سیستم موقع خاموش کردن کمک می‌کند.

اگر می‌خواهید از اینترنت در chroot استفاده کنید باید فایل کانفیگ سرویس دهنده DNS خود را

کپی کنید:

```
# cp -L /etc/resolv.conf etc/resolv.conf
# cp -L /etc/hosts etc/hosts
```

برای شناسایی تنظیمات شل خودتان به chroot به صورت زیر عمل کنید:



```
# chroot ./bin/bash
```

اگر به خطای 'chroot: cannot run command '/bin/bash': Exec format error' برخوردید این احتمال وجود دارد که سیستم مهمان و میزبان با هم از نظر معماری مطابقت نداشته باشند.

## دسترسی به گراب

برای دسترسی به گراب در کنار محیط chroot باید فایل /etc/mtab سیستم را به روز کنید:

```
# grep -v rootfs /proc/mounts > /etc/mtab
```

اگر از bash استفاده می‌کنید می‌توانید پروفایل bash را در chroot تغییر دهید، معمولاً فایل پیکربندی پروفایل bash در یکی از دو مسیر زیر است با این حال می‌توانید با باز کردن فایل زیر از مسیر فایل‌های پیکربندی پروفایل‌ها مطلع شوید:

```
~/bash_profile
```

```
~/bashrc
```

```
# nano /etc/profile
```

## خروج از chroot

وقتی کارتان با chroot تمام شد از محیط chroot خارج شوید:

```
# exit
```

سپس فایل سیستم‌های mount شده را umount کنید:

```
# umount {proc,sys,dev,boot,...}
```

در نهایت هم سعی کنید دیواس‌تان را umount کنید:

```
# cd ..
# umount arch/
```

یکی از دو مورد زیر دلیل مشاهده خطایی مبنی بر مشغول بودن /mnt می‌تواند باشد:

(۱) برنامه دیگری در کنار chroot در حال استفاده از درایو مورد نظر است.

(۲) یک دایرکتوری فرعی هنوز mount شده باقی مانده است. برای مثال شاخه /mnt/arch/usr

در داخل شاخه `/mnt/arch` .

در مورد مثال ذکر شده ابتدا بایستی نقطه فرعی را `umount` کنید؛ برای یادآوری تمام نقاط `mount` شده دستور `mount` را بدون پارامتر صادر کنید:

```
# mount
```

اگر در این مرحله به مشکل برخوردید از تنظیمات اجباری استفاده کنید:

```
# umount -f /mnt
```

بعد از این می‌توانید با خیال راحت سیستم را شروع مجدد کنید.

## معرفی تعدادی برنامه کاربردی و محبوب

بعد از نصب محیط مورد نظر دیگر تقریباً کار شما تمام شده و سیستم شما آماده استفاده است. بهتر است جاوا، پلاگین فلش به همراه یک سری از کدک‌های صوتی و تصویری را برای مشاهده مدیاهای مختلف و ... نصب کنید. در اینجا به پک‌من دستور نصب تعدادی از این برنامه‌ها و ابزارها را می‌دهیم:

```
# pacman -S mplayer openjdk6 dvd+rw-tools libdvdread libdvdcss
# pacman -S ntfs-3g dosfstools gstreamer0.10-bad-plugins dvd+rw-tools
# pacman -S gstreamer0.10-base gstreamer0.10-base-plugins
# pacman -S gstreamer0.10-ffmpeg gstreamer0.10-good
# pacman -S gstreamer0.10-good-plugins gstreamer0.10-ugly
# pacman -S gecko-mediaplayer xine-lib xine-ui flashplugin
```

در ادامه تعدادی از برنامه‌ها که می‌توانید در آرچ لینوکس نصب و استفاده کنید را نام برده‌ایم. برای دیدن لیست کاملی از برنامه‌های عمومی مدخل ویکی آن را مشاهده کنید.

**Rsync**: برنامه‌ای برای پشتیبان‌گیری.

**Deluge**: برنامه‌ای برای کار با کلاینت بیت‌تورنت.

**Empathy**: برنامه‌ای برای چت (گنوم).

**Kopete**: برنامه‌ای برای چت (KDE).

**Pidgin**: برنامه بسیار محبوب پیحین برای چت.

**Evolution**: کلاینت ایمیل.

- Thunderbird**: کلاینت ایمیل شرکت موزیلا.
- Firefox**: مرورگر وب محبوب و معروف فایرفاکس.
- Chromium**: مرورگر محبوب گوگل.
- Opera**: مرورگر پر قدرت وب.
- Hotot**: کلاینت میکرو بلاگینگ.
- Choqok**: کلاینت میکرو بلاگینگ (KDE).
- Filezilla**: برنامه‌ای برای کار با کلاینت FTP.
- GParted**: برنامه‌ای قدرتمند برای پارتیشن بندی، تغییر و ساخت پارتیشن‌های مختلف.
- Okular**: برنامه‌ای برای مشاهده اسناد با فرمت‌های مختلف (KDE).
- Foxit Reader**: برنامه‌ای بسیار سبک برای خواندن PDF.
- goldendict**: دیکشنری قدرتمند برای گنو/لینوکس با قابلیت خواندن لغت‌نامه‌های بایبلون.
- Amarok**: برنامه پخش مولتی مدیا (KDE).
- Rhythmbox**: برنامه پخش مولتی مدیا (گنوم).
- Banshee**: برنامه پخش مولتی مدیا.
- Clementine**: برنامه بسیار قوی برای پخش مولتی مدیا.
- Blender**: برنامه کارهای گرافیکی.
- Gimp**: برنامه بسیار محبوب و قدرتمند گیمپ برای کار بر روی تصاویر.
- Inkscape**: برنامه طراحی و کار بر روی تصاویر.
- Krita**: برنامه کار بر روی تصاویر.
- GalaPix**: برنامه‌ای برای مرور تصاویر.
- XnView**: یک برنامه مرور و کانورت تصویر.
- Mplayer**: برنامه قدرتمند پخش فایل‌های ویدیویی.
- SMPlayer**: یک ظاهر Qt برای Mplayer.
- VLC**: برنامه پر قدرت و محبوب برای پخش فرمت‌های مختلف مولتی مدیا.
- PiTiVi**: برنامه ویرایش ویدیو.
- Audacity**: برنامه قدرتمند ویرایش فایل‌های صوتی.
- Pulseaudio**: برنامه‌ای برای مدیریت سیستم صوت سیستم عامل.
- Taskwarrior**: برنامه برای یادداشت‌های روزانه و یادآوری کارهای روزمره (todo list).
- LibreOffice**: مجموعه قدرتمند اداری.
- Abiword**: یک برنامه پردازش کلمه پر ویژگی.

**LibreOffice Writer**: برنامه پردازش متن حرفه‌ای (این کتاب با همین برنامه نوشته شده است).

**LibreOffice Calc**: برنامه صفحه گسترده.

**lptables**: یک فایروال قدرتمند.

**Brasero**: برنامه رایت CD/DVD (گنوم).

**K3B**: برنامه حرفه‌ای رایت CD/DVD با ویژگی‌های فراوان (KDE).

**Glipper**: برنامه‌ای برای مدیریت کلیپ‌بورد (گنوم).

**Klipper**: برنامه‌ای برای مدیریت کلیپ‌بورد با ویژگی‌های فراوان (KDE).

**Peazip**: برنامه فشرده‌ساز قدرتمند.

**Ark**: برنامه فشرده‌ساز و کار با فایل‌های فشرده.

**P7zip**: برنامه ساخت آرشیوهای 7-Zip.

**Dolphin**: یک مدیر فایل قدرتمند (KDE).

**Nautilus**: یک مدیر فایل قدرتمند (گنوم).

**PCManFM**: یک برنامه مدیر فایل سبک.

**Cario-Dock**: یک داک زیبا و قدرتمند با ویژگی‌های فراوان.

**Avant Windows Navigator**: یک داک با قابلیت پیکربندی فراوان.

**Docky**: یک داک فوق‌العاده.

**Conky**: برنامه مانیتورینگ سیستم.

**Yakuake**: یک ترمینال بی‌نظیر (KDE).

برنامه‌های فوق از طریق پک‌من یا از مخزن کاربران آرچ و توسط Yaourt قابل نصب‌اند.

پایان...



در کانال تلگرام کارنیل هر روز انگیزه خود را شارژ کنید 😊

<https://telegram.me/karnil>

