

بناء توزيعتك الخاصة:

دليل شامل لـ

ARCHISO، CALAMARES، و HELWAN
LINUX



عن المؤلف

اسمي سعيد محمد عبد الملك، وعرفت باسم "سعيد بدر الدين"، إلا أن الكثيرين في مجتمع التقنية يعرفونني باسم "S.M.A"، وهو اختصار لقناتي على يوتيوب. SMA Coding على الرغم من أن خلفيتي الأكاديمية تكمن في المحاسبة المالية، إلا أن شغفي الحقيقي كان دائماً يكمن في التكنولوجيا. هذا الشغف دفعني باستمرار لاستكشاف عالم الحوسبة والبرمجيات الحرة، ودائماً ما كنت أسأل نفسي: "كيف يعمل هذا؟" و"هل يمكنني بناء شيء أفضل؟".

هذا الفضول المستمر قادني عميقاً إلى عالم أنظمة التشغيل، حيث اكتشفت فلسفة آرش لينكس — فلسفة تتماشى تماماً مع رؤيتي الخاصة: البساطة، والتحكم الكامل، والقدرة على بناء أي شيء من الصفر. من هذا المبدأ، ولدت فكرة Helwan Linux. لم يكن مجرد مشروع تقني، بل كان حلمًا بإنشاء توزيعية تعكس هوية المجتمع العربي، وتقدم تجربة مستخدم سلسلة وشخصية، وتقف كمثال ملهم لما يمكن تحقيقه بالشغف والمعرفة.

على مر السنين، قمت بتوثيق هذه الرحلة على قناتي SMA Coding، ساعياً لمشاركة كل ما تعلمته مع المبرمجين وعشاق لينكس في كل مكان. هذا الكتاب هو نتيجتي لتلك الرحلة: ليس مجرد دليل تقني، بل هو خلاصة سنوات من التجربة والخطأ، والتعلم، والمشاركة. أمل أن تجدوا في هذه الصفحات كل ما تحتاجونه لتحويل أفكاركم إلى واقع — وأن تلهمكم لتلعبوا دوراً فاعلاً في مجتمع المصادر المفتوحة.

إهداء

ثمن هذا الكتاب، في أي مكان في العالم، هو أن تقدم دعاءً خالصاً بالرحمة لوالديّ، بغض النظر عن معتقداتك .
التكلفة الحقيقية لهذا الكتاب هي أن تشاركه على أوسع نطاق ممكن.

إهداء

إلى والدي الحبيب،

إلى والدتي الحبيبة،

إلى أخواتي، نبض قلبي.

إلى أحبائي :

نور عيني، ابني محمد؛

بهجة قلبي، ابني عمر؛

وابنتي الغالية ملك .

أنتم أعظم إلهام في حياتي، وحضوركم يذكرني باستمرار بأهمية بناء شيء يستحق أن يُورث .

هذا الكتاب لكم، عربوناً على أن الشغف والمعرفة يمكن أن يخلقا عالماً أفضل.

إهداء

إلى عمالقة التكنولوجيا الذين شكلوا عالمنا الرقمي:

- دينيس ريتشي وكين تومبسون، الأبوان الروحيان للغة C ونظام Unix ، اللذان وضعوا الأساس لكل ما نستخدمه اليوم.
- ريتشارد ستالمان، فيلسوف البرمجيات الحرة، الذي زرع بذور الحركة.
- لينوس تورفالدس، الذي جعل لينكس حقيقة ووجدنا تحت راية واحدة.
- وإلى كل مطور ومبرمج يعمل خلف الكواليس، ويسهر ليالٍ طويلة لإصلاح الأخطاء وكتابة الأكواد.

إلى أيقونات هذا المجتمع:

- **النينجا (The Ninjas)** المطورون الذين يختبئون خلف شاشة سوداء، ويكتبون الأكواد بصمت وبإتقان.
- **الهكرز (Hackers)** ليسوا مدمرين، بل مستكشفين للأنظمة، يتعقبون نقاط الضعف للفهم والتحسين.
- **المساهمون (Contributors)** أولئك الذين يضيفون قطعاً صغيرة تصبح اللبنات الأساسية للمشاريع الضخمة، من سطر كود واحد إلى ترجمة وثيقة.
- **مستخدمو آرش لينكس (Arch Linux users)** الذين يجسدون فلسفة "افعلها بنفسك (DIY)" في حياتهم ويؤمنون بالتحكم الكامل في أنظمتهم.

هذا الإهداء هو عربون امتنان لمجتمع يؤمن بالعطاء بلا حدود. فلتستمر شعلة المعرفة في الاحتراق ببراعة.

عن الكتاب: خارطة طريق للإبداع الرقمي

هل تخيلت يومًا أن نظام تشغيل لينكس يمكن تصميمه خصيصًا ليناسب ذوقك واحتياجاتك الفريدة؟ هل حلمت بامتلاك توزيعية تحمل بصمتك الشخصية، وتحتوي على الأدوات التي تستخدمها يوميًا، وتلبي احتياجات مجتمعك المحلي أو عالمك المهني؟

إذا كانت هذه الأفكار قد راودتك، فإن هذا الكتاب ليس مجرد دليل، بل هو رفيقك في رحلة تحويل هذا الحلم إلى واقع ملموس. إنه دعوة مفتوحة لكل مبدع، مطور، وعاشق لأنظمة التشغيل للانتقال من عالم الاستهلاك إلى عالم الإنتاج. معًا، سننطلق في رحلة مثيرة تبدأ بالجدور العميقة لنظام لينكس، مرورًا باستكشاف الأسرار التقنية للأدوات القوية التي يستخدمها المحترفون، وصولًا إلى بناء مشروعك الخاص من الصفر.

ماذا ستجد في صفحات هذا الكتاب؟

- يتجاوز هذا الكتاب كونه مجرد مجموعة من التعليمات التقنية؛ إنه خارطة طريق استراتيجية تمكنك من الغوص في أعماق عالم لينكس المفتوح، وتمنحك أدوات للتحكم الكامل. مع كل فصل، ستكتسب مهارات متقدمة تضعك في مقعد القيادة:
- **فهم عميق وفلسفي لأرش لينكس: (Arch Linux)** لن تتعلم فقط كيفية استخدام أرش، بل ستفهم الفلسفة التي بني عليها. ستكتشف مبدأ "التحكم الكامل" وكيف يمكن لهذا الأساس المتين أن يكون لوحة قماشية فارغة لمشروعك، مما يحرك من القيود التي تفرضها التوزيعات الأخرى.
- **إتقان Archiso من الصفر إلى الاحتراف:** هذا هو جوهر الكتاب. ستتحول من مستخدم عادي إلى "مهندس إصدارات". ستتعلم كيفية بناء صور ISO قابلة للإقلاع وتخصيص كل جانب من جوانب النظام المباشر بدقة، بما في ذلك إضافة الحزم، واختيار بيانات سطح المكتب، وكتابة نصوص برمجية مخصصة لأتمتة مهامك.
- **تحكم كامل في Calamares:** ستفهم كيفية دمج واجهة تثبيت رسومية سلسلة واحترافية في توزيعتك. ستتعلم في تخصيص كل تفصيل في Calamares، من تصميم واجهة المستخدم إلى تعديل وحدات التثبيت لضمان تجربة تثبيت فريدة تتماشى تمامًا مع رؤيتك.
- **تحويل الأفكار إلى واقع: بناء توزيعتك الخاصة:** بعد إتقان الأدوات، ستمتلك المعرفة والمهارات اللازمة لتحويل أفكارك المجردة إلى توزيعية لينكس عملية، مستقرة، وجاهزة للاستخدام. ستجاوز مجرد إضافة الحزم لتصل إلى مرحلة بناء نظام شامل ومترابط.
- **الاستلهام من تجربة حلوان لينكس: (Helwan Linux)** سنقدم لك نموذجًا عمليًا وملهمًا من قلب المجتمع العربي. تجربة حلوان لينكس هي شهادة حية على أن الشغف والمعرفة يمكن أن ينتجا مشروعًا قويًا ومفيدًا. من خلال مراجعة هذا المشروع، ستكتسب رؤى عملية حول كيفية تجميع كل القطع معًا، وكيف يمكن لمجتمع صغير بناء شيء ذي تأثير كبير.

لمن هذا الدليل؟

لا يقتصر هذا الكتاب على فئة معينة، بل تم تصميمه بعناية ليكون بمثابة جسر يربط بين مختلف مستويات الخبرة في عالم لينكس:

- **للمبتدئ الطموح:** إذا كنت قد بدأت رحلتك للتو في عالم لينكس، فسيرشدك هذا الكتاب خطوة بخطوة. سنبدأ بشرح المفاهيم المعقدة بأسلوب مبسط، مع التركيز على أمثلة عملية تمكنك من بناء مشروعك الأول بنجاح.
- **للمستخدم المتوسط:** إذا كنت على دراية بأساسيات لينكس وتستخدم توزيعات مثل آرتش أو غيرها، فسيفتح لك هذا الكتاب آفاقاً جديدة للتحكم والتخصيص. سنتنقل من مجرد استخدام النظام إلى فهم كيفية بنائه وتعديله من الداخل.
- **للمطور المتقدم:** إذا كنت تتطلع إلى إنشاء بيئات تطوير جاهزة، أو بناء أدوات مخصصة لأغراض أمنية أو علمية، أو حتى إطلاق توزيعة لينكس لمشروعك التجاري أو المجتمعي، فستجد هنا الموارد والأمثلة المتقدمة التي تحتاجها لتنفيذ أفكارك بأعلى كفاءة.

آرتش لينكس: حجر الزاوية للمشاريع الطموحة

لم يتم اختيار آرتش لينكس كقاعدة لهذا الكتاب بشكل عشوائي؛ بل هو قرار استراتيجي يعتمد على عدة مبادئ أساسية جعلته الخيار الأمثل للمشاريع المخصصة:

- **مبدأ KISS (اجعله بسيطاً، أيها الغبي: Keep It Simple, Stupid - يتبنى آرتش مبدأ "البساطة".** هذا يعني أنه يتجنب التعقيد غير الضروري، مما يجعله شفافاً وسهل الفهم لكيفية عمله من البداية إلى النهاية. أنت تبني نظامك بنفسك وتتحكم في كل جزء منه.
- **التحكم الكامل في التخصيص:** على عكس التوزيعات التي تأتي ببيئات سطح مكتب وبرامج محددة مسبقاً، يمنحك آرتش حرية مطلقة. أنت تقرر ما هو موجود في نظامك، مما يتيح لك بناء نظام خفيف وفعال ومرن يلبي احتياجاتك بدقة.
- **نموذج الإصدار المتجدد (Rolling Release):** تحصل على أحدث إصدارات البرامج على الفور، مما يضمن أن تكون توزيعتك دائماً محدثة وتستفيد من أحدث الميزات وإصلاحات الأمان فور توفرها.
- **وثائق لا مثيل لها ومجتمع تعاوني:** لا يوجد نظام لينكس آخر يمتلك وثائق رسمية (Arch Wiki) بهذا المستوى من التفصيل والدقة. إلى جانب مجتمع واسع وتعاوني، ستجد دائماً المساعدة والمعلومات التي تحتاجها للتغلب على أي تحدٍ.

حلوان لينكس: أكثر من مجرد توزيعة

في قلب هذا الدليل، لا نقدم لك الأدوات فحسب؛ بل نلهمك من خلال مثال حي وعملي. **حلوان لينكس** هي تجربة رائدة لمشروع عربي طموح مبني على فلسفة آرتش لينكس، يهدف إلى توفير تجربة لينكس سلسلة ومخصصة للمستخدمين في المنطقة العربية. من خلال تحليل هذا المشروع، سنتعلم كيفية تطبيق المفاهيم التي سنتعلمها، وكيف يمكن للشغف الجماعي أن يخلق شيئاً ذا تأثير حقيقي. حلوان لينكس ليست مجرد توزيعة؛ إنها شهادة حية على أن الإبداع لا يعرف حدوداً وأن المعرفة يمكن تحويلها إلى مشاريع مجتمعية قوية ومفيدة.

نصائح للحصول على أقصى استفادة من هذه الرحلة

المعرفة النظرية وحدها لا تكفي. للحصول على أقصى استفادة من هذا الكتاب، ندعوك لتبني عقلية الباحث والمجرب:

- **التطبيق العملي هو مفتاحك:** لا تقرأ فقط. قم بإنشاء بيئة افتراضية وابدأ في تطبيق كل خطوة؛ التجربة العملية هي أفضل معلم.
- **آرتش ويكي هو أفضل صديق لك:** لا تتردد في الرجوع إلى آرتش ويكي للحصول على تفاصيل إضافية حول أي حزمة أو أمر.
- **كن فضوليًا ومغامرًا:** قم بتغيير الإعدادات، وأضف برامج جديدة، وحاول فهم ما يحدث خلف الكواليس.
- **المشاركة المجتمعية:** شارك أسئلتك وتجاربك في المنتديات التقنية العربية والدولية. المعرفة تنمو من خلال المشاركة.
- **المراجعة المنتظمة:** عد إلى الفصول السابقة كلما شعرت بالحاجة إلى ترسيخ المفاهيم أو اكتساب فهم أعمق.

نتمنى لك رحلة ممتعة ومثمرة في عالم بناء توزيعات لينكس، ونأمل أن تكون هذه الصفحات نقطة انطلاقك نحو إبداعات

رقمية لا حدود لها.

الفصل الأول: أساسيات آرتش لينكس وبيئة العمل

1.1 نظرة تاريخية على لينكس وصعود آرتش

1.1.1 نشأة نظام لينكس (1991) في أوائل التسعينيات، كان عالم أنظمة التشغيل تهيمن عليه أنظمة يونكس التجارية ونظام-MS DOS من مايكروسوفت. كانت أنظمة يونكس قوية ولكنها باهظة الثمن ومغلقة المصدر، مما حد من إتاحتها للجامعات أو الشركات الكبيرة فقط.

في عام 1991، لم يكن لينوس تورفالدس، وهو طالب علوم حاسوب في جامعة هلسنكي بفنلندا، راضيًا عن نظام التشغيل مينيكس (وهو نسخة صغيرة من يونكس لأغراض تعليمية). قرر أن يكتب نواة (kernel) خاصة به كهواية وأطلقها عبر الإنترنت بترخيص مفتوح المصدر.

في إعلانه العام الأول، كتب تورفالدس: "أعمل على نظام تشغيل (مجاني) كهواية، لن يكون شيئًا كبيرًا أو احترافيًا مثل جنو (GNU)."

لكن سرعان ما انضم مئات المبرمجين من جميع أنحاء العالم لتطوير هذا المشروع الجديد، الذي سُمي "نواة لينكس (Linux)". Kernel عندما تم دمجها مع أدوات مشروع جنو (الذي أسسه ريتشارد ستولمان)، حصلنا على ما نعرفه اليوم بنظام التشغيل جنو/لينكس. (GNU/Linux)

1.1.2 بداية التوزيعات نظرًا لأن لينكس كان مجرد نواة بدون أدوات جاهزة للاستخدام، احتاج الناس إلى طريقة سهلة لتجميع النواة مع البرامج الأساسية وتوزيعها كنظام كامل. هكذا ولدت فكرة "التوزيعة. (Distribution)" من بين أوائل التوزيعات كانت:

- سلاك وير: (1993) (Slackware) واحدة من أقدم التوزيعات، ولا تزال موجودة حتى اليوم.
 - دبيان: (1993) (Debian) ركزت على الاستقرار والمجتمع المنظم. أصبحت فيما بعد الأساس للعديد من التوزيعات مثل أوبونتو.
 - ريد هات لينكس: (1995) (Red Hat Linux) استهدفت الشركات، وأصبحت لاحقًا الأساس لـ RHEL و Fedora.
 - كان لكل توزيع فلسفتها الخاصة: البعض ركز على سهولة الاستخدام (مثل أوبونتو لاحقًا)، والبعض على الاستقرار (مثل دبيان)، والبعض الآخر على التخصيص (مثل جنو).
- 1.1.3 ظهور آرتش لينكس (2002) في أوائل العقد الأول من القرن الحادي والعشرين، لاحظ مطور كندي يُدعى جود فينيت أن معظم التوزيعات الحالية إما معقدة للغاية (مثل جنو)، التي تتطلب بناء النظام من المصدر) أو "مُثقلة" لأنها تأتي مُحملة مسبقًا بإعدادات لا يحتاجها كل مستخدم.

في عام 2002، قرر جود فينيت إنشاء توزيعة جديدة أطلق عليها اسم "آرتش لينكس. (Arch Linux)"

- هدفه: توزيعة بسيطة، خفيفة الوزن، ومرنة.
- شعارها KISS: (-) Keep It Simple, Stupid حافظ عليها بسيطة، يا غبي).
- لم تأت بواجهة رسومية؛ بل بدأت من سطر الأوامر.
- استخدمت مدير حزم جديدًا يسمى pacman، والذي كان سهلًا وعمليًا لإدارة الحزم.

لم يكن آرتش لينكس يستهدف المبتدئين، ولكنه جذب مجتمعًا من المستخدمين المتقدمين الذين أحبوا حرية التخصيص والشفافية الكاملة للنظام.

1.1.4 المجتمع يتولى التطوير - آرون غريفين بعد بضع سنوات من قيادة المشروع، قرر جود فينيت التوقف عن تطوير آرتش لينكس لأسباب شخصية. في عام 2007، تولى آرون غريفين القيادة.

خلال فترة تولي غريفين:

- أصبح تطوير آرتش أكثر تنظيمًا ومدفوعًا بالمجتمع.
 - ازدهرت "ويكي آرتش (Arch Wiki)" وأصبحت واحدة من أهم مصادر المعرفة في عالم لينكس.
 - ظهرت مشاريع مشتقة من آرتش، مثل ArchBang وManjaro.
- اليوم، يُدار آرتش لينكس بواسطة فريق من المطورين الأساسيين ومجتمع ضخم من المساهمين، مما جعله واحدًا من أقوى توزيعات لينكس وأكثرها تأثيرًا.

ملخص هذا القسم:

- بدأ لينكس كهواية لطالب جامعي في عام 1991.
- سرعان ما أصبح نظامًا عالميًا بفضل فلسفة المصدر المفتوح.
- ظهرت التوزيعات لتسهيل استخدامه، ولكل منها فلسفتها الخاصة.
- جاء آرتش لينكس (2002) كحل وسط: ليس معقدًا مثل جنّتو، وليس مقيدًا مثل أوبونتو.
- اليوم، آرتش ليس مجرد توزيع، بل هو مدرسة تعلم المستخدم كيفية عمل النظام من الداخل والخارج.

ما الذي يجعل آرتش لينكس فريدًا؟

عندما نتحدث عن توزيعات لينكس، غالبًا ما تتبادر إلى الذهن مصطلحات مثل "سهولة الاستخدام"، "الاستقرار"، أو "الدعم الفني". ولكن مع آرتش لينكس، الوضع مختلف؛ إنه ليس مجرد نظام تشغيل آخر، بل هو فلسفة كاملة للتفاعل مع أجهزة الكمبيوتر. لفهم ما يميز آرتش، يجب أن ننظر إلى عدة مجالات رئيسية:

1.2.1 البساطة (Simplicity)

الشعار الأساسي لآرتش هو KISS (Keep It Simple, Stupid). ومع ذلك، فإن كلمة "البساطة" هنا لا تعني أن النظام سهل للمبتدئين. بل تعني:

- تجنب تعقيد التصميم غير الضروري.
- عدم إضافة أدوات أو واجهات غير ضرورية.
- توفير الحد الأدنى فقط وترك الباقي للمستخدم.

على سبيل المثال، تأتي توزيعات مثل أوبونتو مثبتة مسبقًا بواجهة جنوم (GNOME) ومتصفح فايرفوكس (Firefox) ومجموعة ليبر أوفيس (LibreOffice) في المقابل، عندما تقوم بتهيئة آرتش لأول مرة، فإنه يمنحك نظامًا أساسيًا فارغًا تقريبًا. أنت تقرر: ما هي الواجهة الرسومية التي تريدها؟ أي متصفح؟ أي أدوات مكتبية؟ هذا يجعل آرتش أشبه بقطع الليجو: أنت تبني النظام قطعة قطعة وفقًا لرغباتك.

1.2.2 الشفافية (Transparency)

آرتش لا يخفي شيئًا عن المستخدم:

- جميع ملفات التكوين موجودة كنصوص قابلة للقراءة والتحرير.
- لا يوجد "سحر" خلف الكواليس.
- حتى عملية التثبيت نفسها هي سلسلة من الأوامر التي يكتبها المستخدم، مما يسمح له بفهم كيفية بناء نظام التشغيل خطوة بخطوة.

على سبيل المثال، في أوبونتو، يمكن للمستخدم إضافة مستودع برامج عبر واجهة رسومية أو أمر بسيط مثل `add-apt-repository` في آرتش، يتم ذلك يدويًا عن طريق تحرير الملف `/etc/pacman.conf`، مما يجعلك ترى وتفهم بالضبط ما يحدث.

1.2.3 المرونة (Flexibility)

آرتش لا يفرض عليك أي قرارات:

- هل تريد استخدام KDE Plasma ؟ يمكنك تثبيته.
 - هل تفضل GNOME أو XFCE ؟ كلاهما ممكن.
 - لا تريد واجهة رسومية على الإطلاق وتفضل مدير نوافذ بسيط مثل i3wm أو bspwm ؟ كل ذلك متاح بالكامل.
- هذه المرونة تجعل آرتش مناسبًا لمختلف المستخدمين، من عشاق السرعة والبصمة الصغيرة إلى المتحمسين للأشكال والواجهات المتقدمة.

1.2.4 الإصدار المستمر (Rolling Release)

واحدة من أكبر الاختلافات بين آرتش والتوزيعات الأخرى هو نظام التحديث:

- معظم التوزيعات (مثل أوبونتو أو فيدورا) تصدر إصدارات جديدة كل 6 أشهر إلى سنة. يحتاج المستخدم إلى الترقية بين الإصدارات.
- آرتش، ومع ذلك، هو "إصدار مستمر (Rolling Release)"، مما يعني أن التوزيعة محدثة دائمًا. التثبيت مرة واحدة لآرتش يكفي لسنوات؛ تحتاج فقط إلى إبقائها محدثة عبر `sudo pacman -Syu`.

المزايا:

- تحصل دائمًا على أحدث إصدارات البرامج والنواة.
- لا تحتاج إلى إعادة تثبيت النظام.

العيوب:

- قد يتسبب تحديث غير متوافق في بعض الأحيان في حدوث مشاكل مفاجئة.
- يحتاج المستخدم إلى الانتباه إلى أخبار آرتش (Arch News) قبل التحديث.

1.2.5 مدير الحزم Pacman

أحد ركائز آرتش لينكس هو مدير الحزم `pacman`.

- مصمم ليكون بسيطًا وسريعًا.
- يتعامل تلقائيًا مع التبعية (dependencies).
- الأوامر موحدة ويسهل تذكرها.

أمثلة شائعة:

Bash

##تحديث النظام

sudo pacman -Syu

##تنصيب برنامج

sudo pacman -S firefox

##إزالة برنامج مع ملفاته الإضافية

sudo pacman -Rns firefox

##البحث عن برنامج

pacman -Ss vlc

مقارنة:

- apt في دبيان/أوبونتو قد يتطلب أوامر أكثر تعقيداً.

- dnf في فيدورا أبداً أحياناً في الأداء.

Arch Wiki 1.2.6 موسوعة المعرفة

واحدة من أهم الأشياء التي تميز آرتش ليست التوزيعة نفسها، بل وثائقها.

- تعتبر "ويكي آرتش (Arch Wiki)" واحدة من أكبر وأشمل مراجع لينكس على الإطلاق.

- حتى مستخدمو التوزيعات الأخرى (دبيان، فيدورا، مانجارو) يعتمدون عليها لحل المشكلات.

- تشرح كل شيء، من تثبيت بطاقة رسومات إلى إعداد خادم ويب كامل.

1.2.7 AUR (Arch User Repository)

لدى آرتش مستودعات رسمية ضخمة، لكن قوته الحقيقية تظهر مع: AUR

- هو مستودع تم بناؤه بالكامل بواسطة المجتمع.

- يحتوي على آلاف الحزم التي لن تجدها في المستودعات الرسمية.

- يتم إدارته عبر ملفات PKGBUILD التي تسمح لك ببناء الحزمة على جهازك.
- على سبيل المثال، يمكن العثور بسهولة على برنامج غير متاح رسميًا، مثل جوجل كروم (Google Chrome) ، في AUR عبر أداة yay: مثل

Bash

yay -S google-chrome

ولكن يجب أن تكون حذرًا:

- ليست كل الحزم في AUR مضمونة الجودة العالية أو الأمان.
 - يوصى دائمًا بقراءة ملف PKGBUILD قبل التنصيب.
- 1.2.8 آرتش كأساس لتوزيعات أخرى
- قوة آرتش جعلته أساسًا لعدد من التوزيعات المشتقة:
- مانجارو (Manjaro) تقدم آرتش مع واجهة رسومية جاهزة للاستخدام للمبتدئين.
 - إنديفوروس أو إس (EndeavourOS) توفر تجربة أقرب لآرتش ولكن مع تثبيت أسهل.
 - جارودا لينكس (Garuda Linux) تركز على الرسومات والأداء.
 - حلوان لينكس (Helwan Linux) مثالنا: (توزيعة مصرية مبنية على آرتش بنكهة محلية وهوية فريدة.
- هذا يثبت أن آرتش ليس مجرد نظام تشغيل، بل هو منصة لبناء أنظمة أخرى.

ملخص القسم 1.2:

ما يميز آرتش ليس فقط التكنولوجيا التي بُني عليها، بل الفلسفة التي يتبناها:

- البساطة،
- الشفافية،
- المرونة،
- الإصدار المستمر (Rolling Release) ،
- قوة المجتمع،
- والاعتماد على المستخدم كعنصر أساسي في بناء تجربته.

1.2 الفلسفة العميقة لآرتش لينكس

1.3

آرتش لينكس ليس مجرد توزيع خفيفة أو مرنة؛ بل هو مدرسة كاملة في فلسفة تصميم أنظمة التشغيل. لفهم "روح" آرتش، يجب أن نتعمق في المبادئ التي توجهه.

1.3.1 مبدأ KISS أبقتها بسيطة، أيها الغبي

كلمة "بساطة" هنا لا تعني سهولة الاستخدام. آرتش لا يحاول أن يكون سهلاً للمبتدئين مثل أوبونتو أو منت. بل يعني الوضوح وغياب التعقيد: لا توجد طبقات خفية من البرامج الوسيطة (middleware) أو أدوات إدارة تلقائية تفرض نفسها عليك.

• على سبيل المثال:

○ في أوبونتو، عندما تقوم ب تثبيت تعريف لبطاقة الرسومات، توجد أدوات رسومية مخصصة تقوم بالمهمة نيابة عنك.

○ في آرتش، يتم ذلك عن طريق تثبيت الحزم المناسبة يدويًا من pacman أو مستودع AUR وتعديل ملفات الإعدادات المحددة.

النتيجة: النظام أبسط من الداخل، لكنه يتطلب المزيد من المعرفة والخبرة من المستخدم.

1.3.2 التحكم الكامل (محورية المستخدم)

يبنى آرتش لينكس على فكرة أن المستخدم هو الأدرى باحتياجاته.

• النظام لا يفرض عليك حزمًا أو إعدادات معينة.

• حتى عملية التثبيت لا توفر واجهة رسومية، بل تمنحك الأدوات الأساسية لبناء النظام بنفسك.

هذا التحكم الكامل يجعل آرتش مثاليًا للمطورين والمهندسين الذين يحتاجون إلى بيئة عمل مخصصة. فمثلاً، في أوبونتو، يتم تثبيت النظام مع واجهة GNOME بشكل افتراضي. أما في آرتش، فبعد التثبيت الأولي، لا توجد لديك حتى واجهة رسومية؛ بل شاشة سوداء (TTY). إذا أردت GNOME، تقوم بتثبيتها بنفسك؛ وإذا أردت KDE أو XFCE أو حتى لا شيء، فالقرار لك وحدك.

1.3.3 الشفافية

أحد المبادئ الأساسية في آرتش هو أن كل شيء يجب أن يكون واضحًا ومفهوماً.

• لا يوجد "سحر" يحدث في الخلفية.

• جميع الإعدادات يمكن تعديلها عبر ملفات نصية بسيطة.

• التوثيق (Arch Wiki) يشرح كل خطوة بالتفصيل.

على سبيل المثال، إذا أردت تشغيل خدمة في آرتش، فإنك تستخدم `systemctl enable...` وتفهم ما يحدث في الخلفية. في حين أنك في توزيعات أخرى قد تضغط على زر في واجهة رسومية ولا تعرف ما الذي تم خلف الكواليس.

1.3.4 التعلم بالممارسة

يختلف آرتش لينكس عن معظم التوزيعات لأنه يعلمك أثناء استخدامه:

- عملية التثبيت نفسها هي درس عملي في كيفية عمل نظام لينكس.
 - إعدادات الشبكة، المستخدمين، ومدير الإقلاع (boot manager) هي كلها خطوات يمر بها المستخدم ويفهمها.
 - بمرور الوقت، يتحول مستخدم آرتش من مجرد "مستهلك" للنظام إلى "متحكم" فيه.
- يصف بعض المستخدمين التجربة بأنها: "آرتش لا يعطيك سمكة، بل يعلمك كيف تصطاد".

1.3.5 نموذج التحديث المستمر كجزء من الفلسفة

لم يكن اختيار نظام التحديث المستمر (Rolling Release) لآرتش قرارًا عشوائيًا، بل نابعًا من فلسفته:

- لماذا يجبر المستخدم على إعادة تثبيت النظام كل نة أشهر أو سنة، كما في أوبونتو؟
 - بدلاً من ذلك، دع المستخدم يحصل على أحدث البرامج فورًا، ويستمر نظامه في التطور معه.
- هذا يعكس إيمان آرتش بفكرة أن النظام يجب أن يكون حيًا دائمًا، لا يشيخ أو يصبح قديمًا.

1.3.6 التطور المجتمعي

لا يسعى آرتش لإرضاء الشركات أو التوجه التجاري. بل على العكس:

- المجتمع هو القلب النابض للتوزيع.
 - معظم الحلول تجدها في Arch Wiki ، الذي كتبه المستخدمون.
 - مستودع AUR مبني بالكامل على مساهمات المجتمع.
- يعكس هذا مبدأً فلسفيًا هامًا: المعرفة جماعية وليست حكرًا على مؤسسة أو شركة.

1.3.7 آرتش ليس للجميع (فلسفة انتقائية)

لم يهدف مؤسس آرتش إلى أن تكون التوزيعة سهلة أو مناسبة للجميع.

- إذا كنت مبتدئًا تمامًا، قد تجد آرتش صعبًا جدًا.
 - لكن إذا أردت فهم لينكس من الداخل والتحكم به، ستجد آرتش هو الخيار الأفضل.
- هذا المبدأ جعل آرتش يُعرف أحيانًا بأنه توزيعة "نخبوية"، ليس بمعنى المتعالي، بل بمعنى أنه يتطلب مستوى معينًا من الجدية وحب الاستطلاع.

❏ خلاصة القسم 1.3: يمكن تلخيص فلسفة آرتش لينكس في:

- بساطة التصميم دون تعقيد غير ضروري.
- التحكم الكامل بالنظام.
- الشفافية في كل شيء.
- التعلم بالممارسة.
- التحديث المستمر الذي يواكب العصر.
- مجتمع قوي يدفع عجلة التطوير والدعم.

هذه الفلسفة جعلت من آرتش أكثر من مجرد توزيع: إنه طريقة تفكير وتعامل مع أنظمة التشغيل.

1.4 نموذج التحديث المستمر (Rolling Release)

إحدى أبرز ميزات آرتش لينكس، وما يجعله مختلفاً عن غالبية التوزيعات الأخرى، هي نظام التحديث المستمر، المعروف باسم Rolling Release. هذا المفهوم ليس مجرد طريقة لتوزيع البرامج؛ بل هو جزء أساسي من فلسفة آرتش.

1.4.1 ما هو التحديث المستمر (Rolling Release) ؟

في عالم البرامج، توجد طريقتان رئيسيتان لتوزيع الإصدارات:

• الإصدار الثابت: (Fixed Release)

- تقوم التوزيعة بإصدار نسخة جديدة على فترات زمنية ثابتة (مثل أوبونتو كل 6 أشهر، أو ديبان كل سنتين).
- البرامج داخل هذا الإصدار تبقى بشكل عام كما هي، باستثناء التحديثات الأمنية.
- مثال: أوبونتو 22.04 سيستمر في استخدام نفس إصدار GNOME والنواة حتى إصدار 24.04.

• الإصدار المستمر: (Rolling Release)

- النظام لا يمتلك "إصدارات رئيسية"، بل يتم تحديث الحزم بشكل مستمر.
- عندما يتم إصدار نسخة جديدة من النواة أو أي برنامج، يتم تحديثها فوراً في المستودعات.
- مثال: آرتش لينكس يمتلك دائماً أحدث إصدار من نواة لينكس، دون انتظار "إصدار جديد من آرتش".

1.4.2 مميزات التحديث المستمر

- دائماً أحدث البرامج: تحصل على أحدث إصدار من محرر النصوص المفضل لديك أو بيئة البرمجة بمجرد صدورهما. هذا مناسب جداً للمطورين الذين يحتاجون إلى بيئة حديثة باستمرار.
- لا حاجة لإعادة التثبيت: مع التوزيعات ذات الإصدارات الثابتة، قد تضطر إلى إعادة تثبيت النظام أو ترقيته من حين لآخر. أما مع آرتش، فعملية تثبيت واحدة تكفي لسنوات؛ كل ما عليك هو الاستمرار في التحديث.
- نظام دائم الحيوية: آرتش لا يشيخ أبداً. طالما أنك تقوم بتحديثه، فإنه سيظل دائماً في أحدث صورة له.

1.4.3 عيوب التحديث المستمر

- احتمالية حدوث أعطال: قد يتسبب تحديث غير متوافق أو وجود خطأ في إحدى الحزم في مشكلة. على سبيل المثال، قد يتسبب تحديث لتعريف بطاقة الرسومات أحياناً في مشاكل في الإقلاع.
- مسؤولية أكبر على المستخدم: يجب عليك متابعة أخبار آرتش باستمرار للتأكد من وجود أي تنبيهات مهمة قبل التحديث، وأن تكون مستعداً لإصلاح المشكلات بنفسك.

- استهلاك أكبر للإنترنت: أنت تقوم دائمًا بتنزيل إصدارات جديدة من البرامج.

1.4.4 كيفية تعامل آرتش مع التحديث المستمر

على الرغم من العيوب، يمتلك آرتش نظامًا قويًا لتقليل المشاكل:

- فحص الحزم: قبل أن تدخل أي حزمة إلى المستودع الرسمي، يتم فحصها في مستودع مخصص للتجربة. [testing]
- التوثيق: يتم توثيق أي مشكلة كبيرة فورًا على Arch Wiki أو على صفحة الأخبار.
- دعم المجتمع: يشارك المستخدمون الحلول بسرعة على المنتديات أو موقع Reddit.

1.4.5 استراتيجيات لتحديث آمن

- التحديث بانتظام: عدم التحديث لفترات طويلة يمكن أن يجعل ترقية النظام صعبة بسبب تعارضات كثيرة. من الأفضل التحديث أسبوعيًا أو كل أسبوعين.
- استخدام Timeshift أو النسخ الاحتياطية: يمكنك أخذ نسخة احتياطية قبل التحديث للعودة إليها في حالة حدوث مشكلة.
- قراءة الأخبار قبل التحديث: يبدو أمر sudo pacman -Syu بسيطًا، لكن يجب أن تكون على دراية بما قد يتغير بعده.

1.4.6 مقارنة مع توزيعات أخرى

فيدورا (شبه مستمر)	أوبونتو/ديبيان (ثابت)	آرتش (مستمر)	المعيار
تحديثات سريعة نسبيًا	مستقر طوال فترة الإصدار	أحدث إصدار دائمًا	تحديثات النواة
متوسط	أعلى (مناسب للخوادم)	أقل (لكنه مرن)	استقرار النظام
6~ أشهر	6 أشهر أو أكثر	لا يوجد (مستمر)	الفاصل بين الإصدارات
متوسط	أسهل للمبتدئين	يحتاج إلى متابعة مستمرة	سهولة الإدارة

1.4.7 أمثلة عملية

- مطور ألعاب: يحتاج إلى أحدث مكتبات Vulkan و Mesa → آرتش هو الأنسب.
- شركة خوادم: تحتاج إلى استقرار طويل الأمد بدون مفاجآت → ديبان أو RHEL أفضل.
- مستخدم عادي: يريد جهازه يعمل دائمًا دون قلق → مانجارو (مبني على آرتش ولكنه أكثر استقرارًا).

٧ خلاصة القسم 1.4: نموذج التحديث المستمر في آرتش لينكس هو ميزة قوية تجعله دائماً عصرياً ومواكباً. لكنه سلاح ذو حدين: يمنحك الحرية والحدثة، ولكنه يتطلب مسؤولية ويقتطع.

1.5 مدير الحزم Pacman

أحد الأعمدة الأساسية في آرتش لينكس، وما يميزه عن التوزيعات الأخرى، هو مدير الحزم Pacman. في أنظمة لينكس، مدير الحزم هو الأداة التي تسمح لك بتنزيل البرامج وتثبيتها وتحديثها وإزالتها من مستودعات التوزيع. لكن Pacman يبرز بفضل فلسفته البسيطة وأدائه القوي.

1.5.1 ما هو Pacman ؟

- اسم "Pacman" هو اختصار لـ (Package Manager مدير الحزم).
- هو الأداة الرسمية لإدارة الحزم في آرتش لينكس.
- مكتوب بلغة C ليكون سريعًا وخفيفًا.
- يدير الحزم التي تأتي بامتداد (pkg.tar.zst ملفات مضغوطة تحتوي على البرامج).

1.5.2 مميزات Pacman

- أوامر بسيطة:
 - الأوامر قصيرة وسهلة التذكر.
 - أمثلة S-: للتثبيت، R- للإزالة، Q- للاستعلام.
- إدارة تلقائية للتبعيات:
 - إذا احتاج برنامج لمكتبات إضافية، يقوم Pacman بتثبيتها تلقائيًا.
- السرعة:
 - بفضل تصميمه بلغة C ونظام مستودعات يعتمد على الملفات المضغوطة.
- التوحيد:
 - تُستخدم نفس الأداة لكل شيء (التثبيت، التحديث، البحث، الإزالة).

1.5.3 أوامر Pacman الأساسية

الأمـر	الوظيفة	مثال
pacman -S package	تثبيت برنامج	pacman -S firefox
pacman -R package	إزالة برنامج	pacman -R vlc
pacman -Rns package	إزالة برنامج + التبعيات غير المستخدمة	pacman -Rns gimp
pacman -Ss keyword	البحث عن برنامج في المستودعات	pacman -Ss vlc
pacman -Qs keyword	البحث عن برنامج مثبت محلياً	pacman -Qs python
pacman -Qi package	عرض معلومات عن برنامج مثبت	pacman -Qi nano
pacman -Syu	تحديث النظام بالكامل	pacman -Syu

1.5.4 ملفات إعدادات Pacman

- الملف الرئيسي `/etc/pacman.conf` :
 - يحتوي على إعدادات مثل المستودعات المفعلة، خيارات التثبيت، وإدارة التوقيعات الرقمية.
- قاعدة البيانات المحلية `/var/lib/pacman` :
 - حيث يخزن Pacman معلومات عن الحزم المثبتة.

مثال من pacman.conf:

[options]

HoldPkg = pacman glibc

Architecture = auto

CheckSpace

SigLevel = Required DatabaseOptional

[core]

Include = /etc/pacman.d/mirrorlist

[extra]

Include = /etc/pacman.d/mirrorlist

[community]

Include = /etc/pacman.d/mirrorlist

1.5.5 التوقيعات الرقمية (Package Signing)

- يستخدم Pacman نظام GPG للتحقق من صحة الحزم.
- هذا يعني أن كل حزمة لها توقيع رقمي لضمان عدم تغييرها أثناء النقل.
- في حالة حدوث خطأ في المصادقة، لن يسمح Pacman بالثبيت إلا إذا أجبرت النظام (وهو أمر غير مستحسن).

1.5.6 مقارنة Pacman بمديري حزم آخرين

الميزة	Pacman (Arch)	APT (Debian)	DNF (Fedora)
لغة البرمجة	C (سريع جدًا)	C++	Python/C
صعوبة الأوامر	سهلة وبسيطة	متوسط	متوسط
التبعيات	إدارة قوية	قوية	قوية
التحديثات	دائمًا مستمر (Rolling)	ثابت حسب الإصدارات	نصف سنوية
السرعة	أداء أسرع	أبطأ نسبيًا	أبطأ من Pacman

1.5.7 استخدام Pacman مع مستودعات إضافية

يمكنك إضافة مستودعات إضافية عن طريق تعديل ملف `/etc/pacman.conf`.

مثال: إضافة مستودع `multilib` لتشغيل برامج 32 بت:

[multilib]

Include = `/etc/pacman.d/mirrorlist`

ثم قم بتشغيل `sudo pacman -Syu` :

1.5.8 مشاكل شائعة مع Pacman وحلولها

- قاعدة بيانات تالفة:
 - استخدم `sudo pacman -Syy` لإجبار تحديث قاعدة البيانات.
- ملفات متعارضة أثناء التحديث:
 - الحل: قم بإزالة الملف يدويًا أو استخدم خيار `--overwrite`.
- انقطاع الإنترنت أثناء التحديث:
 - يمكن استئناف العملية بسهولة بعد إعادة الاتصال.

1.5.9 العلاقة بين Pacman وAUR

بينما Pacman قوي جدًا، إلا أنه لا يدير (Arch User Repository) AUR مباشرة.

- بالنسبة لحزم AUR ، تستخدم أدوات مساعدة مثل yay أو paru.
- هذه الأدوات تعتمد في النهاية على Pacman ولكنها تضيف خطوة إنشاء الحزمة من ملف PKGBUILD.

❗ خلاصة القسم Pacman 1.5: ليس مجرد مدير حزم عادي؛ إنه قلب تجربة آرتش لينكس. بساطته وسرعته ومرونته تجعله واحدًا من أسرع وأقوى مديري الحزم في عالم لينكس.

1.6 مستودع مستخدمي آرتش (AUR)

تُعدّ قوة آرتش لينكس الكبيرة في مستودع AUR ، والذي يعتبر أحد أضخم المستودعات المجتمعية في عالم لينكس.

1.6.1 ما هو AUR ؟

- AUR هو اختصار لـ Arch User Repository مستودع مستخدمي آرتش.
 - إنه مستودع ضخم يحتوي على "وصفات" الحزم (PKGBUILDs) مكتوبة بواسطة المجتمع.
 - هدفه هو تمكين المستخدمين من تثبيت البرامج غير المتوفرة في المستودعات الرسمية بسهولة.
- المستودعات الرسمية تحتوي فقط على البرامج التي يختبرها فريق آرتش رسميًا. أما AUR ، فهو مساحة مفتوحة حيث يمكن للمستخدمين مشاركة أي برنامج أو أداة أو حتى سمة.

1.6.2 ما هو ملف PKGBUILD ؟

- ملف PKGBUILD هو برنامج نصي مكتوب بلغة Bash.
- يحتوي على تعليمات لبناء حزمة من المصدر أو من ملفات مجمعة مسبقًا.
- Pacman لا يتعامل مع AUR مباشرة؛ يقوم المستخدم ببناء الحزم بنفسه باستخدام هذا الملف.

مثال بسيط:

```
Bash
pkgname=hello
pkgver=1.0
pkgrel=1
arch=('x86_64')
source=("http://example.com/$pkgname-$pkgver.tar.gz")
md5sums=('SKIP')
build()
{
    cd "$srcdir/$pkgname-$pkgver"
    ./configure --prefix=/usr
```

```

make
}
package()
{
    cd "$srcdir/$pkgname-$pkgver"
    make DESTDIR="$pkgdir/" install
}

```

1.6.3 لماذا يعتبر AUR مهماً؟

- تغطية واسعة: أي برنامج يخطر ببالك غالباً ما تجده في AUR.
- مجتمع ضخم: آلاف المساهمين يرفعون ويحدثون الحزم يومياً.
- مرونة: يمكنك تعديل ملف PKGBUILD بنفسك قبل البناء.
- سرعة التوفر: غالباً ما تُرفع البرامج إلى AUR قبل وصولها إلى المستودعات الرسمية (إن وصلت أبداً).

1.6.4 كيف يعمل AUR عملياً؟

1. تبحث عن الحزمة على موقع <https://aur.archlinux.org> AUR:
2. تنسخ ملف PKGBUILD.
3. تستخدم الأمر `makepkg -si` لبناء الحزمة وتثبيتها محلياً.

1.6.5 أدوات مساعدة لـ AUR

نظرًا لأن التعامل اليدوي مع PKGBUILDs ممل، فقد ابتكر المجتمع أدوات لتبسيط هذه العملية. أشهرها:

الأداة	الميزة
yay	أشهر مساعد؛ يعمل مع Pacman و AUR.
paru	مشابه لـ yay ولكنه بواجهة أبسط.
trizen	يدعم ميزات البحث والبناء المتقدمة.
pamac	واجهة مستخدم رسومية (GUI) تشبه مدير الحزم في مانجارو.

مثال باستخدام yay: yay -S google-chrome هذا الأمر سيجلب في AUR ، ويبني الحزمة، ويقوم بتثبيتها تلقائيًا.

1.6.6 تحديات ومخاطر AUR

- الأمان: بما أن الحزم مكتوبة بواسطة المجتمع، فقد تحتوي على شيفرة خبيثة.
 - الحل: قم دائمًا بفحص ملف PKGBUILD قبل البناء.
- جودة الحزم: ليست كل الحزم في AUR محدثة أو مستقرة.
- الاعتماديات (Dependencies): أحيانًا، توجد اعتماديات غير متوفرة في المستودعات الرسمية.

1.6.7 العلاقة بين AUR والمستودعات الرسمية

- يمكن لحزمة من AUR أن تُعتمد لاحقًا وتُنقل إلى المستودعات الرسمية.
- مثال: العديد من البرامج بدأت حياتها في AUR وأصبحت رسمية بعد اكتسابها شعبية.

1.6.8 مقارنة AUR مع مستودعات مشابهة في توزيعات أخرى

مقارنة بـ AUR	النظام المعادل	التوزيعة
مشابه، لكنه أصغر بكثير حجمًا ومحتوى.	PPA (Personal Package Archives)	Debian
مشابه جدًا، لكنه أقل انتشارًا.	COPR	Fedora
نظام شامل ولكنه أكثر تعقيدًا من AUR.	OBS (Open Build Service)	openSUSE

1.6.9 أمثلة لحزم AUR الشهيرة

- google-chrome: غير متوفر رسميًا بسبب الترخيص.
- spotify: مغلق المصدر، ولكنه متاح عبر AUR.
- visual-studio-code-bin: النسخة الرسمية المجمعة مسبقًا من Microsoft VSCode.
- whatsapp-nativefier: لتحويل WhatsApp إلى تطبيق سطح مكتب.

❗ خلاصة القسم 1.6: يُعدّ AUR الركيزة الثانية بعد Pacman التي تجعل من آر تيش لينكس نظامًا فريدًا. بفضل هذا المستودع، يحصل المستخدم على إمكانية الوصول إلى آلاف البرامج الإضافية غير الرسمية، مما يفتح الباب لحرية ومرونة هائلة، ولكنه يتطلب مسؤولية وفحصًا دقيقًا من المستخدم.

1.7 فلسفة آرتش لينكس: مبدأ (Keep It Simple, Stupid) KISS

أحد أهم ركائز آرتش لينكس هو فلسفة KISS ، التي تقف اختصاراً لـ "Keep It Simple, Stupid": ابقها بسيطة، أيها الغبي. (على الرغم من أن العبارة قد تبدو ساخرة في البداية، إلا أنها تحمل رؤية عميقة لبناء أنظمة البرمجيات: البساطة أقوى من التعقيد.

1.7.1 ما الذي تعنيه البساطة في آرتش؟

البساطة لا تعني نقص الميزات. بل تعني أن النظام مبني على مكونات صغيرة، واضحة، وقابلة للفهم. آرتش لا يحاول إخفاء التعقيد عن المستخدم (كما تفعل توزيعات مثل أوبونتو أو فيدورا)؛ بل يضع الأدوات أمامك ويمنحك التحكم الكامل.

1.7.2 تجليات مبدأ KISS في آرتش لينكس

- ملفات الإعدادات النصية: لا توجد أدوات رسومية معقدة لتغيير الإعدادات. كل شيء تقريباً يتم عبر ملفات نصية قابلة للقراءة والفهم (مثل `systemd`، و `pacman.conf`).
- إدارة الحزم: (Pacman) أداة واحدة قوية تتولى التنصيب والتحديث والإزالة. لا توجد عشرات الأدوات المختلفة كما في بعض التوزيعات الأخرى.
- التركيز على النواة: يمنحك آرتش نظاماً أساسياً نظيفاً. والباقي متروك لك لبنائه خطوة بخطوة.
- التوثيق: (Arch Wiki) بدلاً من تطوير أدوات رسومية تخفي التفاصيل، يركز فريق آرتش على توثيق كل شيء بوضوح.

1.7.3 لماذا يعتبر KISS مهماً للمستخدم؟

- التحكم الكامل: أنت تعرف بالضبط ما هو موجود في نظامك.
- سهولة الصيانة: أي مشكلة يمكن تتبعها بسهولة لأن كل شيء واضح ومباشر.
- المرونة: يصبح النظام مثل صندوق الأدوات الذي تبني منه فقط ما تحتاجه.
- التعلم: مبدأ KISS يجعل آرتش منصة تعليمية ممتازة لفهم لينكس.

1.7.4 الفرق بين KISS والتبسيط الزائف

بعض التوزيعات تحاول أن تكون "سهلة" عبر بناء طبقات رسومية تخفي التعقيد. هذا يؤدي إلى "تبسيط زائف"، حيث يفقد المستخدم القدرة على التحكم الكامل بالنظام، وعندما يحدث خطأ، يصبح من الصعب إصلاحه.

آرتش، على الجانب الآخر، يتبع KISS: يبقي النظام بسيطاً ولكنه شفاف.

1.7.5 أمثلة عملية لمبدأ KISS في آرتش

- تثبيت آرتش: لا يوجد "مثبت رسومي" معقد. أنت تختار الأقراص، والتقسيم، والنواة، وبينه سطح المكتب بنفسك.
- إعداد الشبكة: بدلاً من أداة رسومية ضخمة، يمكنك الاعتماد على أدوات بسيطة مثل ip أو systemd-networkd.
- بناء الحزم: العملية واضحة وبسيطة من خلال ملف PKGBUILD.

1.7.6 انتقادات لمفهوم KISS في آرتش

على الرغم من أن KISS هو نقطة قوة، إلا أن البعض ينتقده:

- منحني تعليمي حاد: قد يجد المبتدئ الأمر صعباً جداً في البداية.
 - عمل متكرر: أحياناً تحتاج إلى إعداد أشياء يدوية كان من الممكن أن تكون مؤتمتة.
 - وقت مستهلك: تخصيص آرتش يستغرق وقتاً أطول من التوزيعات "الجاهزة" مثل أوبونتو.
- ومع ذلك، هذه الانتقادات هي في الواقع جزء من فلسفة آرتش: إذا كنت تريد الراحة المطلقة، فربما آرتش ليس لك.

KISS 1.7.7 على لسان مؤسس آرتش

قال مؤسس آرتش، جود فينت، في مقابلة قديمة: "آرتش ليس للجميع. إنه للمستخدم الذي يريد أن يتعلم ويتحكم، وليس لمن يريد كل شيء جاهزاً".

هذا يلخص الفكرة: آرتش = حرية + بساطة + مسؤولية.

📌 خلاصة القسم 1.7: فلسفة KISS هي ما يجعل آرتش لينكس فريداً بين التوزيعات. إنها فلسفة بساطة واضحة تمنح المستخدم المرونة والتحكم الكامل على حساب منحني تعليمي أكثر حدة. آرتش لا يعد بالسهولة المطلقة، ولكنه يعد بالوضوح والشفافية.

1.8مجتمع آرتش وArch Wiki

لا يمكنك الحديث عن آرتش لينكس دون ذكر المجتمع الذي يقف خلفه. على الرغم من أن آرتش بدأ كتوزيعة صغيرة أسسها جود فينت في عام 2002، فإن سر قوته المستمرة اليوم هو المجتمع النشط الذي يقوم بصيانتها وتطويرها.

1.8.1قوة مجتمع آرتش

- آرتش لينكس ليس مشروعًا تابعًا لشركة؛ بل هو مشروع مجتمعي بالكامل.
- يساهم آلاف المطورين والمستخدمين يوميًا من خلال:
 - إصلاح الأخطاء.
 - تحديث الحزم.
 - إضافة التوثيق.
 - دعم المستخدمين الجدد.
- يعمل المجتمع بشفافية تامة: جميع المناقشات مفتوحة، وجميع القرارات متاحة على القوائم البريدية والمنتديات.

1.8.2 Arch Wiki موسوعة لينكس الأولى

- Arch Wiki هو التوثيق الرسمي للمشروع.
- بدأ كصفحات بسيطة تشرح كيفية تثبيت آرتش، ولكنه اليوم يعتبر أكبر وأشمل مصدر توثيق لأنظمة لينكس بشكل عام، حتى أن مستخدمي التوزيعات الأخرى يعتمدون عليه.

ما الذي يجعل Arch Wiki مميزًا؟

- التفصيل: كل خطوة مشروحة بوضوح.
- التحديث المستمر: أي تغيير في الحزم أو النظام يتم عكسه بسرعة في الويكي.
- الشمولية: لا يقتصر على آرتش؛ بل يضم معلومات عن النظام ككل (النواة، systemd، الشبكات، Xorg، Wayland، إلخ).
- مجاني ومفتوح: مفتوح للجميع، ويمكن لأي مستخدم أن يساهم فيه.

1.8.3 المجتمع كمصدر للدعم

- المنتديات: (Arch Forums) مكان لتبادل الأسئلة والمشاكل والحلول.
- قنوات IRC و Matrix للتواصل المباشر مع المطورين والمستخدمين المخضرمين.
- (Arch User Repository) AUR مستودع ضخم أنشأه المجتمع، يحتوي على مئات الآلاف من الحزم غير الموجودة في المستودعات الرسمية.

1.8.4 روح المشاركة والتعلم

- في آرتش، الفلسفة ليست فقط KISS ، بل أيضًا "عَلِّمْ غيرك."
- المستخدم الجديد الذي يستفيد من المنتدى أو الويكي غالبًا ما يعود لاحقًا لمساعدة الآخرين.
- هذه الدورة المستمرة من التعلم والمشاركة جعلت آرتش أكثر من مجرد توزيع: إنها مدرسة في لينكس.

1.8.5 انتقادات للمجتمع

- يُوصف المجتمع أحيانًا بأنه صارم مع المبتدئين.
- بعض الردود قد تكون قاسية إذا لم يقرأ المستخدم التوثيق أولاً.
- لكن السبب هو أن فلسفة آرتش مبنية على الاعتماد على الذات والقراءة قبل طلب المساعدة.

1.8.6 دروس مستفادة من مجتمع آرتش

- الشفافية تخلق الاستدامة: لا توجد أسرار في التطوير.
- المعرفة المشتركة أقوى من الخبرة الفردية.
- الويكي أفضل من أي أداة رسومية معقدة.

📌 خلاصة القسم 1.8: المجتمع هو القلب النابض لآرتش لينكس، و Arch Wiki هو دماغه. بدون المجتمع، ما كان لآرتش أن يبقى قويًا ومرنًا لأكثر من عقدين من الزمن. ولأي مستخدم جديد، أول شيء يجب أن يتعلمه هو كيفية قراءة التوثيق، وكيفية السؤال بوضوح، وكيفية مشاركة خبرته مع الآخرين.

1.9.1 آرتش لينكس كأساس لتوزيعات أخرى

إحدى أهم علامات نجاح أي توزيعية هي قدرتها على أن تصبح أساسًا لتوزيعات أخرى تبني فوقها. بفضل بساطتها، وحزمها الحديثة، ومستودعاتها القوية، أصبح آرتش لينكس قاعدة لعشرات التوزيعات المشتقة.

1.9.1.1 مانجارو لينكس: آرتش للمبتدئين

ظهرت مانجارو عام 2011 من فريق ألماني بهدف جعل آرتش في متناول المبتدئين. فبينما يركز آرتش على التثبيت اليدوي وإدارة النظام بخطوات دقيقة، توفر مانجارو:

- مثبتًا رسوميًا سهلًا.
 - إعدادات جاهزة للتعريفات (خاصة لبطاقات NVIDIA).
 - مستودعات خاصة بها تقوم بتجميد الحزم لفترة قصيرة لاختبارها قبل طرحها، مما يوفر استقرارًا أكبر.
- تعتبر مانجارو "بوابة" إلى عالم آرتش، حيث تمنح المستخدم تجربة قريبة من آرتش ولكن مع راحة إضافية.

1.9.2 إنديفور أو إس: روح المجتمع

تعود خلفية إنديفور أو إس إلى مشروع Antergos (2002–2019)، الذي كان يهدف إلى تقديم آرتش مع مثبت رسومي وتجربة جاهزة. بعد توقف Antergos، وُلد مشروع EndeavourOS عام 2019 كمبادرة مجتمعية.

تختلف عن مانجارو في أنها تحاول أن تكون أقرب إلى آرتش "الخام" ولكنها توفر:

- مثبتًا رسوميًا.
 - خيارات متعددة لبيئات سطح المكتب (KDE، XFCE، GNOME، إلخ).
 - دعمًا مجتمعيًا قويًا يشبه روح آرتش نفسها.
- تُعتبر خيارًا مثاليًا للمستخدم الذي يريد تجربة مطابقة تقريبًا لآرتش، ولكن مع بداية أسهل.

1.9.3 حلوان لينكس: الهوية المصرية في عالم آرتش

حلوان لينكس هي توزيعية مبنية على آرتش، أنشئت بهدف الجمع بين قوة آرتش وهويته البسيطة مع الرغبة في خلق تجربة محلية مميزة.

مميزات حلوان لينكس:

- تعتمد مباشرة على مستودعات آرتش، مع إضافة تحسينات وتجارب محلية.
- واجهات وأدوات تسهل التعامل مع Pacman، مثل أداة hpm (Helwan Package Manager).

- تركيز على البساطة وهوية بصرية واضحة، مما يمنحها طابعًا مختلفًا عن مجرد استنساخ آخر لآرتش.
- يعكس حلوان لينكس فكرة أن آرتش ليس مجرد توزيع، بل هو منصة مفتوحة تسمح لأي مطور أو فريق بإنشاء مشروع جديد فوقها.

1.9.4 دروس من التوزيعات المشتقة

لا يُقاس نجاح آرتش بمستخدميه المباشرين فقط، بل بمدى انتشار "أبنائه". مانجارو، وإنديفور أو إس، وحلوان لينكس هي أمثلة على كيفية تكييف آرتش ليناسب شرائح مختلفة من المستخدمين:

- المبتدئ الذي يريد السهولة.
- المستخدم المتوسط الذي يريد المرونة مع بداية أسرع.
- المجتمعات المحلية التي تريد هويتها الخاصة.

❏ خلاصة القسم 1.9: لم يعد آرتش لينكس مجرد توزيع قائمة بذاتها؛ بل أصبح نواة لعائلة واسعة من التوزيعات. هذه التوزيعات تثبت قوة الفلسفة الكامنة وراء آرتش وقابليتها للبناء، لتلبية احتياجات جمهور متنوع حول العالم.

1.10 حالات الاستخدام

على الرغم من أن آرتش لينكس هو توزيع للأغراض العامة يمكن لأي شخص تثبيتها، إلا أن طبيعته تجعله أكثر ملاءمة لفئات معينة من المستخدمين. نستعرض هنا أبرز حالات الاستخدام:

1.10.1 للمطورين: الوصول إلى أحدث الحزم

- يستخدم آرتش لينكس نموذج التحديث المستمر (Rolling Release)، مما يعني أن الحزم يتم تحديثها باستمرار.
 - هذا يجعله بيئة مثالية للمطورين الذين يحتاجون إلى:
 - أحدث إصدارات لغات البرمجة (Python، Go، Rust، إلخ).
 - المكتبات وأدوات التطوير الحديثة دون الحاجة إلى انتظار "إصدار جديد" للتوزيع.
 - القدرة على بناء بيئة تطوير مخصصة بالكامل بناءً على احتياجات مشروعهم.
- مثال: مطور يعمل على مشروع ذكاء اصطناعي سيستفيد من توفر أحدث إصدارات TensorFlow أو PyTorch في مستودعات آرتش أو مستودع AUR.

1.10.2 للباحثين والأكاديميين: التحكم الكامل بالبيئة

- غالبًا ما يحتاج الباحثون إلى بيئة تجريبية قابلة للتخصيص بالكامل.
 - يوفر آرتش:
 - القدرة على تثبيت نواة مخصصة أو إصدار معين من المكتبات.
 - التحكم الدقيق بالإصدارات، مما يساعد في إعادة إنتاج النتائج العلمية.
 - مرونة استخدام الأدوات من مستودع AUR أو حتى بناء الحزم يدويًا.
- هذا الأمر مهم بشكل خاص في المجالات العلمية مثل الحوسبة عالية الأداء (HPC)، والفيزياء، والرياضيات التطبيقية.

1.10.3 للمستخدم العادي: بين الفضول والتحدي

- ليس آرتش التوزيع الأنسب للمبتدئين أو المستخدمين الذين يبحثون عن تجربة "فقط تعمل (just works)".
- ومع ذلك، هو خيار ممتاز إذا كان المستخدم:
 - يريد تعلم كيفية عمل لينكس من الداخل.
 - يفضل التحكم الكامل على الإعدادات المسبقة.

○ يمتلك الوقت والفضول لخوض رحلة تعليمية عبر التثبيت اليدوي وإدارة النظام.

قد يجد المستخدم الذي يريد بيئة جاهزة وسريعة دون عناء، توزيعات مثل أوبونتو، أو فيدورا، أو حتى مانجارو أكثر ملاءمة.

❏ خلاصة القسم 1.10: آرتش لينكس هو الملعب المثالي للمطورين والباحثين الذين يحتاجون إلى الحدائثة والمرونة. أما المستخدم العادي فيمكنه الاستفادة من التجربة كرحلة تعليمية، لكنها قد لا تكون الخيار الأفضل لبيئة العمل اليومية ما لم يكن شغوفًا باستكشاف أعماق لينكس.

1.11 خاتمة الفصل الأول

بعد هذه الجولة في عالم آرتش لينكس، يمكننا أن نستنتج أن آرتش ليس مجرد توزيعة أخرى بين عشرات التوزيعات؛ بل هو منهج وفلسفة شاملة.

- فلسفته مبنية على البساطة، الشفافية، والتحكم الكامل.
- التعامل معه يجعلك أقرب إلى نواة لينكس ويساعدك على فهم تركيبته الداخلية، بدلاً من الاكتفاء بالقشرة الخارجية التي توفرها التوزيعات الأخرى.
- كل خطوة في استخدام آرتش هي رحلة تعليمية: من التثبيت اليدوي إلى إدارة الحزم عبر pacman، وحتى تخصيص سطح المكتب حسب رغبتك.

يمكن القول إن آرتش لينكس أشبه بـ "ورشة عمل تعليمية" مفتوحة للمستخدم:

- إذا كنت مطوراً، ستجد أحدث الحزم والأدوات في متناول يدك.
 - إذا كنت باحثاً، ستكون لديك القدرة على التحكم في كل تفصيلة صغيرة وكبيرة داخل بينتك.
 - وإذا كنت مستخدماً عادياً شغوفاً بالتعلم، سيمنحك آرتش فرصة لفهم لينكس بعمق لا توفره أي توزيعة أخرى.
- وبالتالي، يصبح تعلم آرتش بوابة لفهم أعمق للينكس نفسه، ويمنحك ثقة أكبر في التعامل مع أنظمة التشغيل مفتوحة المصدر.
- بانتهاؤ هذا الفصل التمهيدي، نكون قد وضعنا الأساس النظري لفهم آرتش لينكس. الآن، حان الوقت للانتقال إلى الفصل الثاني: إعداد بيئة بناء Archiso، حيث سنبدأ الجانب العملي ونكتشف كيف يمكننا إعداد بيئة العمل لبناء توزيعة مخصصة بدءاً من آرتش.

الفصل الثاني: إعداد بيئة البناء باستخدام Archiso

2.1 مقدمة إلى Archiso

ما هو Archiso ؟ Archiso هو الإطار الرسمي الذي طوره مجتمع Arch Linux وصانوه لبناء صور ISO قابلة للإقلاع (Live ISO images). تستخدم هذه الصور أغراضًا متنوعة، بما في ذلك توزيعية النظام، صيانة الأنظمة، والاختبار. بشكل أساسي، هو "باني صور ISO" الخاص بآرتش، ويسمح لك بـ:

- بناء نسخة افتراضية من آرتش لينكس مطابقة للإصدار الرسمي.
 - أو، الذهاب أبعد من ذلك وإنشاء توزيعية مخصصة بالكامل بهوية جديدة، مثل حلوان لينكس.
- يُبنى Archiso على مبدأ البساطة: فبدلاً من الاعتماد على أدوات معقدة أو أنظمة خارجية، يتكون من مجموعة من السكريبتات وملفات الإعدادات التي يمكنك من توليد نظام تشغيل قابل للإقلاع بصيغة ISO.
- لماذا نستخدم Archiso ؟ لفهم أهميته، دعنا نسأل: "ما الذي يحدد وجود توزيعية لينكس؟" الإجابة تكمن في ملف ISO متاح بسهولة، يمكن للمستخدمين تنزيله وحرقه على محرك أقراص USB والقيام بالإقلاع منه Archiso. هي الأداة التي يمكنك من إنشاء هذا الملف بنفسك.

تتضمن الأسباب الرئيسية لاستخدامه:

- التحكم الكامل: أنت من يحدد الحزم، والإعدادات، وبيئة سطح المكتب.
 - توزيعية شخصية: قم ببناء نظامك الشخصي وشاركه مع الآخرين.
 - أغراض الصيانة: أنشئ ملف ISO مخصصاً لفريقك، مزوداً بأدوات متخصصة في الشبكات أو الأمان.
 - الاستقرار والاختبار: اختبر نسخة من نظامك في بيئة افتراضية مثل VirtualBox قبل توزيعها على نطاق أوسع.
- يمكن القول إن Archiso هو "المسبك" الذي تخرج منه جميع المشاريع المبنية على آرتش.

الفرق بين بناء تثبيت آرتش بسيط وتخصيص توزيعية مثل حلوان لينكس

لتوضيح الأمر أكثر، دعنا نقارن بين سيناريوهين:

العنصر	بناء تثبيت آرتش بسيط	تخصيص توزيعية مثل حلوان لينكس
الهدف	نسخة طبق الأصل من آرتش الرسمي	نسخة مخصصة بهوية جديدة
الحزم المثبتة	الحزم الأساسية فقط	بيئات سطح المكتب + أدوات خاصة بحلوان
الهوية البصرية	شعار آرتش، خلفيات افتراضية	شعار حلوان، سمات مخصصة
الجمهور المستهدف	مستخدمو آرتش المتمرسون	المستخدمون الجدد + المحترفون
حجم ملف ISO	معتدل (700 – 900 ميجابايت)	متغير: خفيف جداً (Fluxbox) إلى ثقيل نسبياً (Cinnamon)

تُبرز هذه المقارنة أن Archiso ليس مجرد أداة نسخ، بل هو آلية قوية لصناعة "تجربة مستخدم" مميزة.

المتطلبات الأساسية للبناء باستخدام Archiso

قبل الشروع في أي عملية بناء، تأكد من استيفاء المتطلبات التالية:

- أجهزة مناسبة:
- معالج حديث (يفضل x86_64، مع دعم virtualization إذا كنت ستختبر على QEMU/VirtualBox).
- ذاكرة وصول عشوائي (RAM) لا تقل عن 4 جيجابايت (ويوصى بـ 8 جيجابايت).
- مساحة تخزين 20 جيجابايت أو أكثر (تتطلب عملية البناء ذاكرة مؤقتة ودلائل مؤقتة).
- يوصى بشدة بقرص SSD لتسريع عمليات الضغط وفك الضغط.
- نظام آرتش لينكس أو نظام مشتق منه:
- يجب أن يكون لديك نظام آرتش لينكس مثبت، أو توزيع مبنية عليه (مثل حلوان لينكس).
- السبب: يعتمد Archiso على مكتبات وأدوات محددة من بيئة آرتش.
- اتصال إنترنت جيد:
- تتضمن عملية البناء تنزيل مئات الحزم من المستودعات.
- قد يؤدي الاتصال الضعيف إلى أخطاء في التوقيع.
- صلاحيات الجذر:
- تتطلب عملية البناء تعديل ملفات أساسية وتثبيت حزم.
- يفضل استخدام sudo بدلاً من تسجيل الدخول كـ root مباشرةً.

ملاحظة تاريخية

Archiso ليس تطوراً حديثاً.

- كانت أصوله بسيطة جداً: سكريبت بسيط لبناء ملف ISO الرسمي لآرتش.
- بمرور الوقت، اكتشف المستخدمون إمكاناته لإنشاء نسخ مخصصة خاصة بهم.
- اليوم، تعود أصول معظم التوزيعات المبنية على آرتش (مثل Manjaro، EndeavourOS، Artix، و حلوان لينكس) إلى Archiso.

مثال عملي لتوضيح المفهوم

تخيل هذا السيناريو:

- أنت تعمل على تثبيت آرتش عادي.
 - قررت بناء نسخة تجريبية خفيفة الوزن تسمى Helwan Light.
 - ببساطة، تقوم بنسخ ملفات الإعدادات من Archiso ، وتعُدّل ملف packages.x86_64، وتزيل بيانات سطح المكتب الثقيلة مثل GNOME أو KDE.
 - في غضون ساعة أو ساعتين، سيكون لديك ملف ISO بحجم 600 ميجابايت يُقلع إلى بيئة Fluxbox خفيفة جدًا.
- هذا يوضح كيف يبسّط Archiso العملية بشكل كبير ويوفر بيئة ملموسة للتجريب.

خاتمة (مقدمة الفصل)

- Archiso هو المحرك الأساسي لبناء أي توزيع مبنية على آرتش.
- إنه يمكنك من إعادة بناء نسخة بسيطة أو الابتكار بتوزيع جديدة تحمل هويتك الفريدة.
- المتطلبات الأساسية ليست صعبة للغاية ولكنها تتطلب أجهزة مناسبة ونظام آرتش جاهز.
- ستوجهك الأقسام القادمة (2.2 – 2.9) خطوة بخطوة، بدءًا من تثبيت الأدوات الأساسية وصولًا إلى اختبار توزيعك في VirtualBox.

2.2 مقدمة إلى Archiso

تتمحور عملية بناء توزيعية مبنية على آرتش لينكس باستخدام أداة archiso حول دليل /releng، والذي يُعد نواة عملية البناء. يحتوي هذا الدليل على جميع المكونات التي تُحدد صورة ISO النهائية، بدءًا من نظام الملفات الجذر (/airootfs) وقوائم الحزم المطلوبة وصولًا إلى ملفات الإقلاع لكل من أنظمة BIOS و UEFI.

يوضح هذا المستند وظيفة كل ملف ومجلد داخل /releng، مع شرح كيفية دمج هذه العناصر لإنشاء بيئة حية جاهزة للاستخدام أو التثبيت. فهم هذا الهيكل يمنح مطور التوزيعية القدرة على تخصيص كل جزء بدقة، من الحزم المضمنة إلى واجهة المستخدم عند الإقلاع الأول.

تثبيت أداة archiso أداة archiso هي المسؤولة عن بناء صورة ISO. إنها متاحة في مستودع extra الرسمي لآرتش لينكس ويمكن تثبيتها بالأمر التالي:

Bash

sudo pacman -S archiso

ملاحظة: يُنصح بالقيام بعملية البناء داخل بيئة آرتش نظيفة (أو على الأقل داخل chroot أو VM) لتجنب التعارضات مع النظام المضيف.

الحصول على ملف إعدادات releng تأتي أداة archiso مع ملفات إعدادات معدة مسبقًا (profiles)، بما في ذلك (releng) نفس الملف الذي يستخدمه آرتش لينكس لإنتاج ملف ISO الرسمي. (النسخ ملف الإعدادات إلى دليل العمل الخاص بك:

Bash

cp -r /usr/share/archiso/configs/releng/ ~/archive/

الآن، لديك نسخة محلية من ملف الإعدادات في ~/archive/ يمكنك تعديلها بحرية دون التأثير على النسخة الأصلية.

archive /

└─ airootfs

| └─ etc

| └─ root

| └─ usr

└─ bootstrap_packages.x86_64

└─ efiboot

└─ grub

└─ packages.x86_64

```
└─ ? pacman.conf
└─ ? profiledef.sh
└─ ? syslinux
    └─ ? archiso_head.cfg
    └─ ? archiso_pxe-linux.cfg
    └─ ? archiso_pxe.cfg
    └─ ? archiso_sys-linux.cfg
    └─ ? archiso_sys.cfg
    └─ ? archiso_tail.cfg
    └─ ? splash.png
    └─ ? syslinux.cfg
```

يُقدم هذا المخطط الشجري نظرة عامة على تخطيط ملف إعدادات **archiso**. في الأقسام التالية، سيتم شرح كل دليل و ملف بالتفصيل.

releng/

هو الدليل الأساسي لملف إعدادات **archiso**. كل ما بداخله يُحدد:

- الحزم التي سيتم تثبيتها في ملف **ISO**.
- إعدادات الإقلاع (**EFI / BIOS**).
- ملفات تخصيص المستخدم و **root**.
- إعدادات **Pacman**.
- ملفات محمل الإقلاع (**grub/syslinux**).

❏ airootfs/

هذا هو الجزء الأكثر أهمية، ويمثل نظام الملفات الجذر (rootfs) الذي سيتم بناؤه داخل ملف الـ ISO. أي ملف يتم وضعه هنا سيظهر تحت /بعد إقلاع النظام من الـ ISO.

- → etc/❏ إعدادات النظام: يحتوي على ملفات الإعداد مثل shadow، passwd، hostname، issue، إلخ. أي تغييرات هنا تنعكس في النظام الحي. (live system)
- → root/❏ دليل المستخدم root داخل البيئة الحية. هنا يمكنك وضع سكريبتات التخصيص مثل customize_airootfs.sh أو ملفات الإعداد الشخصية (dotfiles) مثل .bashrc، .zshrc.
- → usr/❏ ملفات ومكتبات إضافية للمستخدم. مثل إضافات لـ /bin/ و /lib/، أو أي أدوات تريد تثبيتها مسبقًا.

❏ bootstrap_packages.x86_64

قائمة بالحزم الأساسية اللازمة لبناء بيئة ISO الأولية (بيئة bootstrap). هذه الحزم لا تُضمّن في النظام النهائي، ولكنها تُستخدم أثناء عملية البناء للسماح لـ archiso بالدخول إلى بيئة معزولة (chroot) ومتابعة البناء.

❏ efiboot/

يحتوي على ملفات محمل الإقلاع EFI إما systemd-boot أو GRUB. لـ UEFI هذا المجلد يحمل ملفات efi. والإعدادات الخاصة بـ UEFI. الإقلاع.

❏ grub/

إعدادات محمل الإقلاع GRUB إذا اخترت استخدامه). يحتوي على ملفات الإقلاع لقائمة GRUB، والخلفية، وقوالب الإعدادات.

❏ packages.x86_64

هذه هي قائمة الحزم التي ستكون متاحة داخل النظام الحي بعد الإقلاع. إنها أهم ملف لتخصيص ملف الـ ISO الخاص بك. هنا، سنُدرج:

- الأدوات الأساسية (مثل bash، vim، nano).
- بيئات سطح المكتب أو مديري النوافذ. (window managers)
- أدوات الشبكة والصيانة.

pacman.conf

ملف إعدادات مدير الحزم pacman داخل النظام الحي. يمكنك استخدامه لتحديد:

- المستودعات الرسمية.
- مستودعات إضافية (إضافية/مخصصة).
- التحقق من التوقيعات وخيارات أخرى.

profiledef.sh

هو "عقل" ملف الإعدادات، حيث يُعرّف خصائصه الأساسية. يُحدد:

- اسم التوزيعة (iso_name)، (iso_label).
- الإصدار. (iso_version)
- النواة (kernel) التي سيتم استخدامها.
- إعدادات وضع الإقلاع. (UEFI/BIOS)
- دلائل البناء والمراحل المؤقتة. (build and staging directories)
- يتبع archiso هذا الملف خطوة بخطوة لبناء ملف الـ ISO.

syslinux/

يحتوي على ملفات محمل الإقلاع Syslinux ، والذي يُستخدم للإقلاع عبر BIOS/Legacy.

- → archiso_head.cfg بداية إعدادات قائمة الإقلاع (العنوان، المهلة الزمنية).
- → archiso_pxe-linux.cfg & archiso_pxe.cfg إعدادات الإقلاع عبر PXE (الإقلاع عبر الشبكة).
- → archiso_sys-linux.cfg & archiso_sys.cfg إعدادات الإقلاع القياسية من ملف الـ ISO.
- → archiso_tail.cfg نهاية إعدادات قائمة الإقلاع.
- → splash.png خلفية رسومية لشاشة Syslinux.
- → syslinux.cfg الملف الرئيسي الذي يضم ملفات الإعدادات الأخرى (head ، tail ، sys ، pxe) ويُعرّف قائمة الإقلاع.

ملخص:

- `airootfs/` = نظام الملفات الجذر الذي سيتم بناؤه داخل ملف الـ ISO.
- `packages.x86_64` = الحزم التي ستكون في النظام الحي. (live system)
- `bootstrap_packages.x86_64` = الحزم المطلوبة لعملية البناء فقط.
- `pacman.conf` = إعدادات مدير الحزم.
- `profiledef.sh` = خصائص التوزيعة والإصدار.
- `efiboot/ + grub/ + syslinux/` = ملفات الإقلاع لكل وضع.

2.3 ملفات الإعداد الأساسية (Core Configuration Files)

يمثل هذا الفصل حجر الزاوية في رحلتنا لتخصيص توزيع آرنتش لينكس. هنا، سنتعمق في قلب عملية البناء، مستكشفين الملفات التي لا تحدد فقط محتوى ملف الـ ISO النهائي، بل تتحكم أيضًا في كيفية بناء هذا الملف.

فهم هذه الملفات الأربعة: `profiledef.sh` ، `packages.x86_64` ، `bootstrap_packages.x86_64` ، و `pacman.conf` — هو مفتاح تحقيق تحكم كامل بمشروعك، مما يمكنك من بناء توزيع فريدة حقًا.

نحن لا نتحدث فقط عن قوائم الحزم؛ هذه الملفات تمثل "وصفة" متكاملة يتبناها `archiso` خطوة بخطوة لتهيئة النظام. كل سطر في هذه الملفات له غرض محدد، وكل خيار يترجم إلى سلوك معين أثناء عملية البناء أو داخل النظام الحي بعد الإقلاع.

2.3.1 `profiledef.sh`: العقل المدبر وراء توزيعك المخصصة

هذا الملف هو جوهر أي ملف إعدادات لـ `archiso`. إنه سكريبت `Bash` يحتوي على جميع التعريفات والإعدادات التي تحدد هوية صورة الـ ISO الخاصة بك، وسلوك إقلاعها، وعملية بنائها. اعتبره "قائمة المهام" التي يتبناها `archiso` لإنتاج نظامك الحي.

- `#!/usr/bin/env bash`: هذا هو سطر "shebang" المعروف، والذي يوجه النظام لتنفيذ هذا السكريبت باستخدام مترجم `Bash`.

- `shellcheck disable=SC2034`: هذا تعليق خاص بأداة `shellcheck` (أداة تحليل سكريبتات `Bash` هنا، نطلب من `shellcheck` تجاهل تحذير معين يتعلق بالمتغيرات غير المستخدمة (`SC2034`) ، حيث قد تكون بعض المتغيرات مخصصة لأغراض محددة أو مستخدمة داخليًا بواسطة `archiso`.

بيانات التعريف الخاصة بـ (ISO Metadata) ISO

هذه المتغيرات تُعرّف هوية وخصائص صورة الـ ISO النهائية.

- `iso_name="Helwan-Linux-stable"`: اسم ملف الـ ISO الذي سيتم إنشاؤه. في مثالنا، سيكون `Helwan-Linux-stable-x86_64.iso` (يضيف `archiso` تلقائيًا بنية المعالج).
- `iso_label="Helwan-Linux-stable-v1.1"`: تسمية الـ ISO. هذا هو النص الذي يظهر عند تحميل الـ ISO أو الإقلاع منه (على سبيل المثال، في قائمة مدير الإقلاع). يجب أن يكون موجزًا (عادةً بحد أقصى 32 حرفًا، ولكن 16 حرفًا هو الأفضل لتوافق أوسع).
- `iso_publisher="helwanlinux <helwanlinux@gmail.com>"`: تتضمن اسم الجهة المسؤولة عن التوزيع ونقطة اتصال (عنوان بريد إلكتروني في هذا المثال).
- `iso_application="Helwan Linux Live/Rescue DVD"`: يوصف موجز لغرض الـ ISO. يوضح استخدامه المقصود، مثل "Live/Rescue DVD" أو "System Installer".
- `iso_version="v1.1"`: إصدار التوزيع. هذا مهم لتتبع التحديثات وتحديد الإصدارات المحددة.

إعدادات البناء الأساسية (Core Build Settings)

تُعرف هذه المتغيرات بنية وعناصر عملية البناء الأساسية.

- `install_dir="arch":` دليل التثبيت داخل الـ ISO. عندما يقوم المستخدم بتحميل الـ ISO ، سيتم وضع ملفات النظام في هذا المسار. الافتراضي لـ Arch Linux هو `arch`.
- `buildmodes=('iso')` أوضاع البناء المتاحة.
 - `iso`: يشير إلى أننا سنبنّي صورة ISO قابلة للإقلاع.
 - قد تشمل الأوضاع الأخرى `tar` لإنشاء أرشيف لنظام الملفات (أو `vendor` لإنشاء صور مخصصة لأجهزة معينة)، ولكن `iso` هو الأكثر شيوعًا.
- `arch="x86_64":` بنية المعالج المستهدفة. هنا، `x86_64` تعني أن الصورة مصممة لأنظمة 64 بت.
- `pacman_conf="pacman.conf":` اسم ملف إعدادات Pacman. يشير هذا إلى ملف `pacman.conf` الموجود في نفس دليل ملف الإعدادات، والذي سيستخدم لتهيئة مدير الحزم داخل النظام الحي.
- `airootfs_image_type="squashfs":` نوع ضغط نظام الملفات الجذر.
 - `squashfs` هذا هو النوع القياسي للأنظمة الحية (Live Systems) ، ويوفر ضغطًا ممتازًا وسرعة قراءة.
 - قد تشمل الخيارات الأخرى `ext4`، ولكن `squashfs` هو الأكثر استخدامًا لصور الـ ISO.

خيارات وأدوات الضغط (Compression Options and Tools)

تحدد هذه المتغيرات كيفية ضغط نظام الملفات الجذر (`airootfs`) والملفات الأخرى.

- `airootfs_image_tool_options=(-comp 'xz' -Xbcj 'x86' -b '1M' -Xdict-size '1M')`: ضغط `squashfs`.
 - `-comp 'xz'`: يستخدم خوارزمية ضغط `xz`، والتي توفر نسبة ضغط عالية جدًا مقابل وقت معالجة أطول.
 - `-Xbcj 'x86'`: يحدد فلتراً خاصاً (x86 Code Branch) لتطبيق الضغط على شيفرة بنية `x86`، مما يحسن نسبة الضغط.
 - `-b '1M'`: حجم الكتلة (block size) الذي ستستخدمه الأداة. حجم أكبر يعني عادةً ضغطاً أفضل ولكنه يتطلب ذاكرة أكبر أثناء الضغط.
 - `-Xdict-size '1M'`: حجم قاموس الضغط (dictionary size). قاموس أكبر يؤدي بشكل عام إلى ضغط أفضل.
 - `bootstrap_tarball_compression=('zstd' -c -T0 '--auto-threads=logical' --long '-19')`: الضغط للأرشيف المستخدم لبناء البيئة الأولية.
 - هنا، نستخدم `zstd`، وهو ضاغط سريع وفعال جدًا.
- `-c`: يكتب المخرجات إلى المخرج القياسي (`stdout`).

- T0-يستخدم جميع الأنوية المعالجة المتاحة (لتحقيق أقصى سرعة).
- auto-threads=logical--يضبط عدد الخيوط تلقائيًا بناءً على عدد الأنوية المنطقية.
- long--يُمكن وضع "long" ، والذي يزيد من فعالية الضغط (مفيد للملفات الكبيرة).
- 19-مستوى الضغط (19 هو الأعلى، مما يعني أقصى ضغط مقابل وقت أطول).

أوضاع الإقلاع (Boot Modes)

هذا المتغير هو أحد أهم المتغيرات، لأنه يحدد كيف يمكن إقلاع ملف الـ ISO.

bootmodes=('bios.syslinux.mbr'

'bios.syslinux.eltorito'

'uefi-ia32.systemd-boot.esp' 'uefi-x64.systemd-boot.esp'

'uefi-ia32.systemd-boot.eltorito' 'uefi-x64.systemd-boot.eltorito')

يعكس هذا مدى تعقيد وتوافق archiso. دعنا نحلل هذه السلاسل:

• bios.syslinux.mbr:

- bios:يشير إلى وضع إقلاع BIOS القديم).
- syslinux:يستخدم Syslinux كمحمل إقلاع.
- mbr:يستخدم Master Boot Record (MBR) لإنشاء قطاع إقلاع قابل للتنفيذ على القرص، مما يسمح بالإقلاع من الأجهزة القديمة.

• bios.syslinux.eltorito:

- bios:وضع BIOS.
- syslinux:يستخدم كمحمل إقلاع Syslinux.
- eltorito:يستخدم معيار El Torito لإنشاء قرص CD/DVD قابل للإقلاع. هذا هو التنسيق القياسي لصور الـ ISO القابلة للإقلاع.

• uefi-ia32.systemd-boot.esp:

- uefi:يشير إلى وضع إقلاع UEFI (Unified Extensible Firmware Interface).
- ia32:يشير إلى دعم المعالجات 32 بت (x86).
- systemd-boot:يستخدم systemd-boot المعروف سابقًا بـ (gummiboot كمحمل إقلاع UEFI).
- esp:يشير إلى هيكل الملفات داخل الـ ISO الذي يحاكي قسم نظام (EFI System Partition - ESP).

• uefi-x64.systemd-boot.esp:

○ uefi: وضع UEFI.

○ x64: يدعم معالجات 64 بت. (x86_64)

○ systemd-boot: محمل الإقلاع systemd-boot.

○ esp: هيكل ملفات ESP.

• uefi-ia32.systemd-boot.eltorito:

○ uefi: وضع UEFI.

○ ia32: يدعم 32 بت.

○ systemd-boot: محمل الإقلاع systemd-boot.

○ eltorito: يستخدم معيار El Torito لبناء صورة ISO قابلة للإقلاع لـ UEFI.

• uefi-x64.systemd-boot.eltorito:

○ uefi: وضع UEFI.

○ x64: يدعم 64 بت.

○ systemd-boot: محمل الإقلاع systemd-boot.

○ eltorito: معيار El Torito لإنشاء صورة ISO قابلة للإقلاع لـ UEFI.

تطبيق عملي: هذا يعني أن الصورة الناتجة ستكون قادرة على الإقلاع من معظم الأجهزة الحديثة (UEFI) والقديمة (BIOS) ، باستخدام محملات الإقلاع الافتراضية لـ Arch Linux (systemd-boot) لـ UEFI و Syslinux لـ BIOS.

أذونات الملفات (File Permissions)

يحدد هذا القسم الأذونات للملفات والأدلة الهامة داخل نظام الملفات الجذر (airrootfs) لضمان الأمان والتشغيل السليم.

file_permissions=(

["/etc/shadow"]="0:0:400"

["/root"]="0:0:750"

["/root/.automated_script.sh"]="0:0:755"

["/root/.gnupg"]="0:0:700"

["/usr/local/bin/choose-mirror"]="0:0:755"

["/usr/local/bin/Installation_guide"]="0:0:755"

["/usr/local/bin/livecd-sound"]="0:0:755"

["/etc/polkit-1/rules.d"]="0:0:750"

["/etc/sudoers.d"]="0:0:750"

)

هنا، يُستخدم التنسيق ["file_path"]="owner:group:permissions".

- owner:group: المعرف المستخدم (UID) ومعرف المجموعة (GID). 0:0 تعني عادةً المستخدم root والمجموعة root.
- permissions: الأذونات بتنسيق ثماني (octal).
 - 400: للقراءة فقط للمالك. (r-----)
 - 750: للقراءة والكتابة والتنفيذ للمالك (rwxr-x---) ؛ للقراءة والتنفيذ للمجموعة.
 - 755: للقراءة والكتابة والتنفيذ للمالك (rwxr-xr-x) ؛ للقراءة والتنفيذ للآخرين.
 - 700: للتنفيذ فقط للمالك. (rwx-----)

أهمية هذه الأذونات:

- /etc/shadow: يحتوي على كلمات المرور المشفرة. يجب أن يكون مملوكًا لـ root وقابلًا للقراءة بواسطة root فقط (400).
- /root: دليل المستخدم 750. root يعني أن root يمكنه القراءة والكتابة والتنفيذ، بينما يمكن للمجموعة القراءة والتنفيذ فقط (مفيد إذا تم إنشاء مجموعات محددة).
- /root/.automated_script.sh: إذا كان لديك سكريبتات مؤتمتة، فيجب أن تكون قابلة للتنفيذ (755).
- /root/.gnupg: للمجلدات التي تحتوي على مفاتيح GPG ، من الأفضل تقييد الوصول إلى root فقط (700).
- /usr/local/bin/: هذا المسار هو الموقع القياسي لإضافة السكريبتات المخصصة. تعيين أذونات 755 يجعلها قابلة للتنفيذ من قبل الجميع.
- /etc/polkit-1/rules.d و /etc/sudoers.d: تحتوي هذه المجلدات على قواعد تصعيد الامتيازات. تقييد الوصول إليها (750) يمنع المستخدمين غير المصرح لهم من تعديلها.

ملخص الفصل

ملف profiledef.sh هو كنز دفين للتحكم في عملية بناء الـ ISO. لقد رأينا كيف يحدد هذا الملف:

- هوية التوزيع (الاسم، الإصدار، الناشر).
- بنية النظام (المعالج، دليل التثبيت، نوع صورة النظام).
- ميزات الإقلاع (BIOS ، UEFI ، Syslinux ، systemd-boot).

- إعدادات الأمان (أذونات الملفات).

- أدوات الضغط لتحسين حجم الـ ISO وكفاءة البناء.

فهمك العميق لهذا الملف سيمكنك من تخصيص توزيعتك بما يتجاوز مجرد قائمة الحزم، ليصل إلى سلوك النظام الأولي. في الفصول اللاحقة، سنتوسع في هذه التعريفات بينما نتعمق في مجلدات `/airootfs` وملفات الإقلاع.

2.3.2 packages.x86_64: قائمة حزم النظام الحي

إذا كان `profiledef.sh` هو المخطط، فإن `packages.x86_64` هو قائمة المكونات الفعلية للنظام. هذا الملف هو قائمة نصية بسيطة بالحزم التي سيقوم `archiso` بتنصيبها داخل النظام الحي. (Live System)

فلسفة الاختيار

عند بناء توزيعة مخصصة، ففكر بعناية في الحزم التي تدرجها. هناك توازن بين توفير بيئة غنية بالميزات والحفاظ على حجم ملف الـ ISO صغيرًا.

مثال للمحتوى

Core System

`base`

`linux`

`linux-firmware`

`nano`

`vim`

Desktop Environment (Example: GNOME)

`xorg-server`

`gnome`

`gnome-extra`

Networking and Connectivity

`networkmanager`

`dhcpcd`

`wpa_supplicant`

System Utilities

htop

neofetch

pacman-contrib

نصائح لتخصيص القائمة:

- ابدأ بـ **base** و **linux**: هذه الحزم ضرورية لتوفير نظام قابل للإقلاع.
- أضف الأدوات الأساسية: أدوات مثل **nano**، **vim**، و **git** غالبًا ما تكون ضرورية للمستخدمين.
- حدد بيئة سطح المكتب الخاصة بك: اختر بيئة سطح مكتب (مثل **GNOME**، **KDE Plasma**، **XFCE**) أو مدير نوافذ (مثل **i3**، **Sway**) مع جميع الاعتماديات الضرورية.
- ضع في اعتبارك الأجهزة: إذا كانت توزيعك تستهدف أجهزة معينة، ففكر في تضمين التعريفات الخاصة (مثل بطاقات **Wi-Fi** أو بطاقات الرسومات).
- إدارة الحزم: يستخدم **archiso** أداة **pacman**، ولكن يمكنك إضافة أدوات مساعدة مثل **yay** إذا رغبت في ذلك.

2.3.3 bootstrap_packages.x86_64: أساسيات عملية البناء

على عكس **packages.x86_64**، فإن الحزم المدرجة في **bootstrap_packages.x86_64** ليست جزءًا من النظام الحي النهائي. بدلاً من ذلك، هي الحزم التي يحتاجها **archiso** نفسه لإنشاء بيئة بناء معزولة (**chroot**) وتثبيت الحزم الأخرى. هذه الحزم ضرورية للتنفيذ الفعال لعملية التجميع.

مثال لمحتوى **bootstrap_packages.x86_64**

(مثال "Helwan Linux" :

arch-install-scripts

archiso

base

base-devel

linux

linux-firmware

شرح هذه الحزم:

- **arch-install-scripts:** توفر هذه الحزمة أدوات أساسية لتنصيب Arch Linux ، وأهمها الأداة **arch-chroot**. تسمح هذه الأداة لـ **archiso** بالدخول إلى بيئة النظام الذي يتم بناؤه لإجراء عمليات التنصيب والتعديلات.
- **archiso:** هذه هي الأداة نفسها! يجب أن تكون متاحة داخل بيئة البناء.
- **base:** توفر الحد الأدنى من الحزم الضرورية لعمل نظام Arch Linux ، بما في ذلك **systemd** والأدوات الأساسية الأخرى.
- **base-devel:** تتضمن أدوات تطوير البرمجيات الأساسية (مثل **gcc** ، **make**) التي قد يحتاجها **archiso** لبناء مكونات معينة (على الرغم من أن هذا أقل شيوعاً في معظم عمليات بناء الـ **ISO**).
- **linux** و **linux-firmware:** النواة (**kernel**) وبرامجها الثابتة المرتبطة بها، وهي ضرورية لبناء بيئة تشغيل أساسية.

متى يتم تعديل هذا الملف:

في معظم السيناريوهات، لن نحتاج إلى تعديل **bootstrap_packages.x86_64** القائمة الافتراضية كافية لبناء معظم توزيعات Arch Linux. ستحتاج فقط إلى تعديله إذا كنت تقوم ببناء متخصص للغاية أو تواجه مشاكل محددة داخل بيئة البناء نفسها.

2.3.4 ملف pacman.conf: مركز التحكم لعالمك المخصص

ملف `pacman.conf` هو محور التحكم المركزي لـ `pacman` ، وهو مدير الحزم الذي يشكل العمود الفقري لإدارة البرمجيات في Arch Linux. عندما تقوم ببناء صورة ISO باستخدام `archiso` ، فإن هذا الملف يحدد كيفية جلب الحزم وتثبيتها، سواء داخل بيئة النظام الحي (live system) أو خلال مرحلة البناء الأولية (bootstrap). دعنا نفصل هذا الملف سطرًا بسطر، مع التركيز على الخيارات الأكثر أهمية لك كمطور توزيع:

قسم ([options] الإعدادات العامة)

يعمل هذا القسم كلوحة تحكم شاملة لـ `pacman`

المسارات: (Paths)

`#RootDir = /`

`#DBPath = /var/lib/pacman/`

`#CacheDir = /var/cache/pacman/pkg/`

`#LogFile = /var/log/pacman.log`

`#GPGDir = /etc/pacman.d/gnupg/`

`#HookDir = /etc/pacman.d/hooks/`

تحدد هذه المسارات الأماكن التي يخزن فيها `pacman` قواعد بياناته (`DBPath`) ، حيث يحتفظ بالحزم التي تم تنزيلها (`CacheDir`) ، حيث يكتب سجلات العمليات (`LogFile`) ، وحيث يبحث عن مفاتيح `GPG (GPGDir)` والأدوات المساعدة (`HookDir`)

- في سياق `archiso` عندما يعمل `pacman` داخل بيئة `chroot` أثناء عملية البناء، تشير هذه المسارات عادةً إلى مواقع مؤقتة داخل دليل البناء. على سبيل المثال، قد يكون `CacheDir` موجوداً داخل مجلد مؤقت لمنع تلويث ذاكرة التخزين المؤقت للنظام المضيف.

- للتخصيص: نادرًا ما ستحتاج إلى تغيير هذه المسارات عند بناء صورة ISO ، ولكن فهمها ضروري لفهم كيفية عمل `pacman` داخليًا.

`IgnorePkg / IgnoreGroup:` و `HoldPkg`

`HoldPkg = pacman glibc`

`IgnorePkg =`

`IgnoreGroup =`

- `HoldPkg`: يمنع هذا الخيار الحزم المحددة من الترقية التلقائية. هذا أمر حاسم للحفاظ على استقرار النظام، خاصةً للحزم الأساسية مثل `pacman` و `glibc` نفسها.
- `IgnorePkg`: إقائمة بالحزم التي لا تريد أبدًا أن يقوم `pacman` بتحديثها.

- **IgnoreGroup:** قائمة بمجموعات الحزم التي لا تريد تحديثها.
 - في سياق **archiso:** يُعد الحفاظ على **pacman** و **glibc** في **HoldPkg** ممارسة جيدة لضمان عدم تعطل عملية البناء بسبب تغييرات غير متوقعة في هذه الحزم الحيوية.
- Architecture = auto:**
- يحدد هذا الخيار بنية النظام المستهدفة. تعني **auto** أن **pacman** سيحاول اكتشاف البنية تلقائياً (مثل **x86_64**).
- في سياق **archiso:** بما أنك تبني صورة ISO لبنية محددة، يمكنك تعيينها صراحةً هنا (مثلاً **Architecture = x86_64**)، لكن **auto** تعمل جيداً بشكل عام.
- ParallelDownloads = 5:**
- يحدد هذا الخيار عدد الحزم التي يمكن لـ **pacman** تنزيلها بشكل متزامن. يمكن أن تؤدي زيادة هذا العدد (مثلاً إلى 5 أو 10) إلى تسريع عملية التنزيل بشكل كبير، خاصةً مع اتصال إنترنت جيد.
- في سياق **archiso:** سرعة التنزيل الأسرع تعني بناء أسرع لـ **ISO** ، مما يجعل هذا خياراً مفيداً للغاية.
- SigLevel و LocalFileSigLevel:**
- SigLevel = Required DatabaseOptional**
- LocalFileSigLevel = Optional**
- تحدد هذه الإعدادات مستوى التحقق من التوقيعات الرقمية للحزم.
- **SigLevel = Required DatabaseOptional:** هذا يعني أن **pacman** سيطلب توقيعاً صالحاً (**Required**) للحزم التي يتم تنزيلها من المستودعات البعيدة، ولكنه سيقبل الحزم ذات التوقيعات الاختيارية (**DatabaseOptional**) أو حتى الحزم غير الموقعة من قواعد بيانات المستودعات.
 - **LocalFileSigLevel = Optional:** يشير هذا إلى أن الحزم المثبتة من الملفات المحلية (مثل تلك المنسوخة يدوياً) يمكن أن تكون غير موقعة أو موقعة بشكل اختياري.
 - في سياق **archiso:** يضمن هذا الإعداد مستوى معيناً من الأمان عند جلب الحزم مع الحفاظ على مرونة كافية لعدم تعطيل البناء في حالة وجود مشكلة مؤقتة في التوقيع.

قسم ([repositories]) مصادر الحزم

هنا تحدد أين سيبحث **pacman** عن الحزم. الترتيب هنا حاسم، حيث سيستخدم **pacman** أول مستودع يحتوي على الحزمة المطلوبة.

المستودعات الرسمية:

[core]

Include = /etc/pacman.d/mirrorlist

[extra]

Include = /etc/pacman.d/mirrorlist

#[multilib]

#Include = /etc/pacman.d/mirrorlist

- [core], [extra]: هذه هي المستودعات الأساسية في Arch Linux
- Include = /etc/pacman.d/mirrorlist: يخبر هذا pacman باستخدام ملف /etc/pacman.d/mirrorlist لاختيار أفضل مرآة لـ core و extra.
- [multilib]: هذا الخيار معطل افتراضياً. إذا كنت بحاجة إلى دعم للتطبيقات 32-بت على نظام 64-بت، ستحتاج إلى إلغاء التعليق عليه (إزالة #).
- في سياق archiso: تتيح هذه الإعدادات لـ archiso الوصول إلى مستودعات Arch Linux الرسمية لتنزيل جميع الحزم التي قمت بتحميلها في bootstrap_packages.x86_64 و packages.x86_64.

مستودعك المخصص (مثال) helwan :

[helwan]

SigLevel = Optional TrustedOnly

Server = https://helwan-linux.github.io/\$repo/\$arch

هذا القسم هو مثال حي على كيفية إضافة مستودعك المخصص.

- [helwan]: اسم المستودع (يجب أن يكون فريداً).
- SigLevel = Optional TrustedOnly: هنا، أنت تخبر pacman أنه يفضل أن تكون الحزم موقعة (Optional) ، ولكنه سيقبلها إذا كانت تأتي من مستودع موثوق به (TrustedOnly) حتى لو لم تكن التوقيعات الصريحة موجودة. يمكن أن يكون هذا مفيداً للمستودعات المخصصة حيث قد لا تكون كل حزمة موقعة باستمرار.
- Server = [https://helwan-linux.github.io/\\$repo/\\$arch](https://helwan-linux.github.io/$repo/$arch): هذا هو عنوان URL للمستودع.

○ سيتم استبدال \$repo تلقائياً باسم المستودع (أي. helwan).

○ سيتم استبدال \$arch بالبنية المستهدفة (مثلاً. x86_64).

لماذا يهم؟ يتيح لك هذا تضمين حزمك المخصصة، أو حتى الإصدارات المعدلة من حزم Arch ، مباشرة في صورة ISO الخاصة بك. ستحتاج إلى خادم ويب (أو حتى مستودع محلي) لاستضافة هذه الحزم.

المستودع المحلي المخصص (مثال) custom :

#[custom]

#SigLevel = Optional TrustAll

#Server = file:///home/custompkgs

هذا مثال آخر لمستودع مخصص، ولكنه يعتمد على نظام الملفات المحلي.

- **SigLevel = Optional TrustAll** هنا، يعني **TrustAll** أن **pacman** لن يتحقق من أي توقيعات على الإطلاق. لا يُنصح بهذا للمستودعات العامة بسبب المخاوف الأمنية ولكنه يمكن أن يكون مفيداً للتجارب السريعة أو في البيئات المعزولة.
- **Server = file:///home/custompkgs** يشير إلى مسار مجلد على نظام الملفات المحلي يحتوي على الحزم.
- في سياق **archiso** إذا كنت تخزن حزمك المخصصة محلياً داخل ملفات البروفایل، يمكنك استخدام هذا النوع من الإعداد.

مثال عملي توضيحي

لنفترض أنك تبني توزيعية "Helwan Linux" وتريد تضمين حزمة مخصصة اسمها **helwan-tools** غير متوفرة في مستودعات **Arch** الرسمية.

1. أنشئ مستودعك المخصص:

○ أنشئ دليلاً لمستودعك، مثلاً **~/my_helwan_repo**.

○ ضع ملف حزمك (**helwan-tools.pkg.tar.zst** أو أي تنسيق آخر) داخل هذا الدليل.

○ أنشئ قاعدة بيانات المستودع باستخدام **repo-add**:

2. **repo-add ~/my_helwan_repo/helwan.db.tar.gz ~/my_helwan_repo/helwan-tools-*.pkg.tar.zst**

○ ارفع محتويات (**~/my_helwan_repo**) بما في ذلك **helwan.db.tar.gz** و **helwan-tools-*.pkg.tar.zst** إلى خادم ويب، أو استخدم **file://** إذا كان محلياً.

3. عدّل **pacman.conf** إذا قمت برفعه إلى **/x86_64/helwan-linux.github.io/helwan/**، سيبدو قسم **pacman.conf** الخاص بك مثل المثال المقدم:

4. **[helwan]**

5. **SigLevel = Optional TrustedOnly**

6. **Server = https://helwan-linux.github.io/\$repo/\$arch**

7. عدّل **packages.x86_64** أضف حزمك المخصصة إلى القائمة:

8. ... **#حزم أخرى...**

الآن، عندما يقوم archiso ببناء صورة ISO ، سيعرف pacman من أين يجلب helwan-tools وسيقوم بتنصيبه في النظام الحي.

pacman.conf (من profiledef.sh شرح إضافي)

لقد قدمت أيضًا مقطعًا مثالًا من ملف profiledef.sh يوضح استخدام pacman.conf كملف خارجي:

Bash

```
# pacman_conf="pacman.conf" # يشير إلى ملف pacman.conf الموجود في نفس دليل البروفایل
```

يشير هذا السطر في profiledef.sh إلى أن archiso سيستخدم الملف المسمى pacman.conf، الموجود داخل نفس دليل البروفایل (مثل /releng/ أو /archive/ في مثالك)، كملف تكوين لـ pacman. هذا هو السلوك الافتراضي والموصى به، لأنه يحافظ على جميع إعدادات pacman مدمجة مع البروفایل الخاص بك.

ملخص الفصل

ملف pacman.conf هو أداة قوية لتخصيص مصادر الحزم الخاصة بك والتحكم في كيفية جلب الحزم وتنصيبها. من خلال إدارته بعناية، يمكنك ضمان أن نظامك الحي لديه إمكانية الوصول إلى جميع الحزم التي يحتاجها، بما في ذلك الحزم المخصصة، بطريقة آمنة وفعالة. فهم هذه الإعدادات يمنحك تحكمًا كاملاً في "المستودعات" التي تعتمد عليها توزيعتك.

2.4.1 دليل /airootfs/ بعد فهم الملفات الأساسية، ينتقل القارئ إلى كيفية تخصيص نظام الملفات نفسه. يشرح هذا الفصل كيفية إضافة ملفاتك الخاصة أو تعديل إعدادات النظام مباشرة.

- شرح دور دليل /airootfs/ كبيئة لنظام الملفات الجذر.(root filesystem)

- شرح الدلائل الفرعية:

- etc/: إعدادات النظام.

- root/: دليل المستخدم الجذر.

- usr/: ملفات ثنائية إضافية.

2.4.1.1 نظرة داخل دليل /etc/ يحتوي دليل /etc/ على ملفات الإعدادات والتهيئات لمعظم البرامج والنظام نفسه. إنه المكان الذي تقوم فيه بتخصيص سلوك النظام وخدماته.

30-touchpad.conf: xorg.conf.d/* X11 يضبط إعدادات لوحة اللمس (touchpad) لبيئة Xorg الرسومية، مثل حساسية اللمس والتمرير **grub: default *** ملف إعدادات GRUB ، محمل الإقلاع (bootloader) ، الذي يتحكم في كيفية بدء تشغيل النظام وخيارات الإقلاع المتاحة: **group *** يحدد هذا الملف المجموعات الموجودة على النظام وأعضائها: **gshadow *** يحتوي على كلمات المرور المشفرة للمجموعات (إذا تم استخدامها: **hostname *** (يحتوي على اسم المضيف (hostname) لجهازك **lightdm.conf: lightdm *** الإعدادات الرئيسية لـ LightDM ، وهو مدير عرض (display manager) للواجهة الرسومية.

- **greeter.conf: slick-greeter** إعدادات شاشة تسجيل الدخول (greeter) لـ **locale.conf: *** يحدد إعدادات اللغة والمنطقة للنظام، مثل تنسيق التاريخ والوقت والأرقام: **localtime *** يشير إلى الوقت المحلي الذي تم إعداده للنظام **mkinitcpio.conf: *** ملف إعدادات لـ mkinitcpio ، يُستخدم لإنشاء صور (initramfs) نظام ملفات أولي يتم تحميله قبل نظام الملفات الرئيسي: **archiso.conf: *** mkinitcpio.conf.d (ملف إعدادات خاص بـ Archiso ، وهي أداة لإنشاء صور ISO قابلة للإقلاع من: **linux.preset: *** mkinitcpio.d Arch Linux يحدد وحدات النواة (kernel modules) التي يجب تضمينها في صورة **broadcom-wl.conf: *** modprobe.d (يحدد خيارات التحميل لوحدة نواة تعريف شبكة Broadcom اللاسلكية: **motd *** رسالة اليوم: (Message Of The Day) تُعرض للمستخدمين عند تسجيل الدخول: **pacman.conf: *** ملف الإعدادات الأساسي لـ Pacman ، مدير حزم Arch Linux ، يحدد المستودعات (repositories) وإعدادات التحديث: **hooks/uncomment-mirrors.hook: *** pacman.d (خطاف (hook) لـ Pacman يقوم بإلغاء التعليق عن خيارات المرآة (mirror) لإعدادات التحديث.

- **hooks/zzzz99-remove-custom-hooks-from-airootfs.hook: *** مصمم لإزالة الخطافات Pacman المخصصة من: **passwd *** يحدد إعدادات المستخدمين، مثل أسماء المستخدمين، **polkit-1 * rules.d/49-nopasswd_global.rules: *** قواعد لـ PolicyKit ، تسمح بإجراء بعض الإجراءات دون طلب كلمة مرور: **resolv.conf: *** يحدد خوادم DNS التي يستخدمها النظام لحل أسماء النطاقات: **shadow *** (domain names) يحتوي على كلمات مرور المستخدمين المشفرة وبيانات المصادقة الأخرى: **skel: *** دليل "الهيكل" (skeleton) يحتوي على الملفات والإعدادات الافتراضية التي يتم نسخها إلى الدليل الرئيسي للمستخدم الجديد عند إنشائه.

- **bashrc, .bashrc, Xresources** ملفات إعدادات شائعة لبيئة X (مثل إعدادات المظهر) و shell Bash الأوامر والأسماء المستعارة).

- cinnamon: إعدادات وتهيئة خاصة لبيئة سطح المكتب Cinnamon ، بما في ذلك إعدادات التطبيقات المصغرة (applets) والأدوات.
- config/dconf/user: بيانات إعدادات نظام DConf.
- config/nemo/desktop-metadata: بيانات تعريف لسطح المكتب في مدير ملفات Nemo.
- icons/default/index.theme: يحدد أيقونات النظام الافتراضية.
- Templates: قوالب للملفات الجديدة (مثل جداول البيانات، المستندات، السكريبتات-10/ssh * sshd_config.d).
 archiso.conf: إعدادات إضافية لخادم SSH خاصة ببنية: sudoers.d * g_wheel. Archiso. ملف إعدادات لـ
 sudo، يمنح أعضاء مجموعة wheel صلاحيات لتنفيذ الأوامر كمستخدم الجذر * systemd. دليل إعدادات لـ
 systemd، نظام التهيئة (init system) في العديد من توزيعات Linux الحديثة.
- journald.conf.d/volatile-storage.conf: إعدادات للتخزين المؤقت لسجلات journald.
- logind.conf.d/do-not-suspend.conf: يمنع النظام من الدخول في وضع الإسبات (suspend).
- network: إعدادات شبكة systemd، تحدد كيفية تهيئة واجهات الشبكة (Ethernet)، WLAN، WWAN.
- network.conf.d/ipv6-privacy-extensions.conf: تمكين امتدادات الخصوصية لـ IPv6.
- resolved.conf.d/archiso.conf: إعدادات خاصة لـ (systemd-resolved) نظام حل الأسماء لـ Archiso.
- system: يحتوي على ملفات service و target. مختلفة تحدد كيفية تشغيل الخدمات وإقلاع النظام. أمثلة تشمل
 display-manager.service (لبدء الواجهة الرسومية)، (NetworkManager.service لإدارة الشبكة)،
 sshd.service (لخادم SSH، و) reflector.service لتحسين مرآة Pacman.
- system-generators: مولدات النظام (System generators)، مثل systemd-gpt-auto-generator، التي تساعد
 في التهيئة التلقائية لأقسام GPT.
- zram-generator.conf: إعدادات لإنشاء أقسام ZRAM ضغط الذاكرة لتحسين أداء النظام * xdg.
 reflector/reflector.conf: ملف إعدادات لـ reflector، وهي أداة تساعد في العثور على أسرع مرآة لـ Pacman
 وتحديثها.

2.4.1.2 دليل /root المحتويات يُستخدم دليل /root كـ الدليل الرئيسي (Home Directory) للمستخدم الجذر (root) ، الذي يمتلك صلاحيات إدارية كاملة على نظام لينكس. أي تعديلات أو ملفات توضع هنا تخص المستخدم الجذر حصريًا، وعادةً لا تؤثر على المستخدمين الآخرين.

• root/

• .automated_script.sh:

- الوصف: هذا الملف هو عبارة عن سكريبت شل (shell script) يُحتمل أنه مُعد لتنفيذ مهام آلية أو مُسلسلة. النقطة (.) في البداية تشير إلى أنه ملف مخفي.
- التقنية: تُستخدم السكريبتات من هذا النوع لتنشغيل سلسلة من الأوامر تلقائيًا، مما يوفر الوقت والجهد للمستخدم الجذر عند أداء الإجراءات المتكررة. يمكن أن تتضمن أوامر لتنصيب البرامج، تهيئة النظام، إجراء النسخ الاحتياطي، أو عمليات إدارية أخرى.
- الاستخدام: يمكن تشغيل هذا السكريبت يدويًا بواسطة المستخدم الجذر، أو قد يتم استدعاؤه بواسطة عملية أخرى أو حدث نظام محدد.

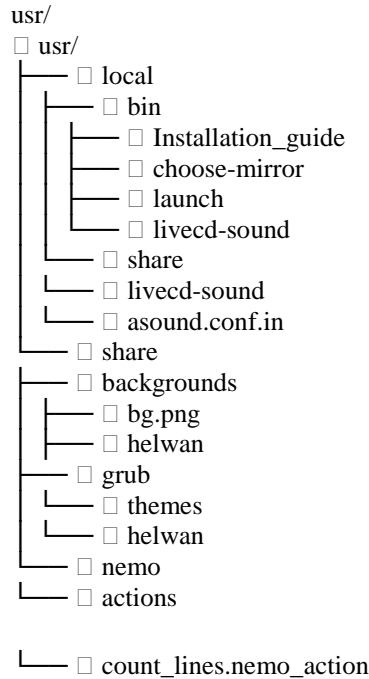
• .gnupg/:

- الوصف: يحتوي هذا الدليل على إعدادات وبيانات GnuPG (GNU Privacy Guard). أداة لتشفير وفك تشفير البيانات والتوقيع الرقمي. النقطة (.) في البداية تجعل الدليل مخفيًا.
- التقنية:
- **sddaemon.conf**: هذا هو ملف إعدادات لـ **sddaemon (Smartcard Daemon)** ، وهو مكون من GnuPG يتعامل مع البطاقات الذكية (smart cards) والأجهزة التشفيرية الأخرى. يحدد هذا الملف كيفية تفاعل sddaemon مع هذه الأجهزة، مثل تحديد مسارات معينة أو خيارات الاتصال.
- الاستخدام: يُستخدم GnuPG لتأمين الاتصالات، التحقق من سلامة البيانات، وإدارة المفاتيح. وجود هذا الدليل في /root يشير إلى أن المستخدم الجذر قد يقوم بإعداد وتنفيذ عمليات تشفير أو توقيع تتطلب إدارة مفاتيح متقدمة.

• .zlogin:

- الوصف: هذا الملف هو سكريبت يتم تنفيذه تلقائيًا عند تسجيل دخول المستخدم الجذر إلى النظام (خاصة عند تسجيل الدخول عبر الطرفية/الواجهة النصية). إنه مشابه لـ **bash_profile** أو **profile**. ولكنه خاص بـ **zsh** (shell) أو الواجهات النصية الأخرى التي تدعمه، وهو شائع في بعض البيئات. النقطة (.) تجعله ملفًا مخفيًا.
- التقنية: عند تسجيل دخول المستخدم، تقرأ الواجهة النصية هذا الملف لتنفيذ الأوامر الموجودة بداخله. يمكن استخدامه لتعيين متغيرات البيئة، تشغيل برامج معينة عند بدء الجلسة، عرض رسائل ترحيب مخصصة، أو تحميل إعدادات خاصة بالمستخدم.
- الاستخدام: يسمح هذا للمستخدم الجذر بتخصيص بيئة عمله مع كل تسجيل دخول، مما يضمن أن أدوات أو إعدادات معينة جاهزة فور بدء الجلسة.

باختصار، في هذا السياق، يحتوي دليل `/root` على أدوات وسكريبتات خاصة بالمستخدم الجذر، سواء لأتمتة المهام (`.automated_script.sh`)، أو لإدارة التشفير والأمان (`.gnupg/scdaemon.conf`)، أو لتخصيص بيئة تسجيل الدخول (`.zlogin`).



2.4.1.3 فهم بنية نظام الملفات - دليل `/usr` بعد استيعاب المفاهيم الأساسية للملفات الضرورية ودور دليل `aiirootfs` كبنية لنظام الملفات الجذر، ينتقل هذا الفصل لاستكشاف أحد أهم الدلائل في أي نظام لينكس: دليل `/usr`. يعمل هذا الدليل كمستودع رئيسي للتطبيقات والبيانات المشتركة التي ليست بالغة الأهمية لعملية إقلاع النظام الأولية، ولكنه حيوي لتشغيل بيئة المستخدم والوظائف الإضافية.

دور دليل `/usr` الاسم `/usr` هو اختصار تقليدي لـ "`Unix System Resources`" (أو "`User System Resources`" موارد نظام يونكس أو موارد نظام المستخدم). على الرغم من أن هذا الاختصار يعود إلى الأيام الأولى لأنظمة يونكس، إلا أن وظيفته الحالية هي احتواء الملفات التنفيذية (`executables`)، المكتبات (`libraries`)، ملفات الرأس (`header files`)، والتوثيق (`documentation`) التي تشكل الجزء الأكبر من بيئة المستخدم. ميزة رئيسية لدليل `/usr` هي أن محتوياته للقراءة فقط (`read-only`) في معظم الحالات. هذا يعني أن النظام يمكن أن يعمل حتى لو تم تحميل قسم `/usr` كنظام ملفات للقراءة فقط. يضمن هذا التصميم استقرار النظام ويمنع التغييرات العرضية في البرامج الأساسية.

استكشاف الدلائل الفرعية الرئيسية داخل `/usr` ينقسم دليل `/usr` إلى عدة دلائل فرعية رئيسية، كل منها يخدم غرضًا محددًا:

- `/usr/bin`: يحتوي على الملفات التنفيذية والبرامج الأساسية التي يمكن لجميع المستخدمين تشغيلها. على عكس أوامر الإقلاع الأساسية الموجودة في `/bin`، يحتوي هذا الدليل على معظم التطبيقات والأدوات شائعة الاستخدام.
- `/usr/sbin`: يحتوي على الملفات التنفيذية والبرامج المخصصة للمسؤول (`root`)، مثل أدوات إدارة النظام والشبكة.

- **/usr/lib:** يحتوي على المكتبات الثابتة والمشاركة المطلوبة للملفات التنفيذية في **/usr/bin** و **/usr/sbin**.
- **/usr/local:** هذا هو الموقع المخصص للبرامج التي تم تجميعها من المصدر أو تثبيتها يدويًا بواسطة المسؤول. هذا يمنع التعارض مع الحزم التي يوفرها النظام (الموجودة في **/usr/bin** ، **/usr/lib** ، إلخ).
- **/usr/share:** يحتوي على ملفات بيانات غير قابلة للتنفيذ مشتركة بين البرامج، مثل ملفات التكوين العامة، الأيقونات، الخلفيات، صفحات man ، والتوثيق الآخر.

نظرة تفصيلية على محتويات **/usr** في سياق Archiso في سياق Archiso ، الأداة المستخدمة لإنشاء صور ISO قابلة للإقلاع من Arch Linux ، يعكس دليل **/usr** داخل **airootfs** البنية Arch Linux القياسية، مع بعض التعديلات أو الإضافات الخاصة بالبيئة الحية (live environment) أو أدوات التثبيت. دعنا نلقي نظرة مفصلة على البنية المحددة التي قدمتها:

/usr/local هذا الدليل هو الموقع القياسي لتثبيت البرامج والتطبيقات التي ليست جزءًا من التوزيع الأصلي ولكن تم تثبيتها يدويًا بواسطة المستخدم أو المسؤول، وغالبًا بعد تجميعها من المصدر. يضمن هذا الفصل بين حزم النظام الأساسية التي تتم إدارتها بواسطة مدير حزم (مثل Pacman) والتطبيقات المخصصة، مما يقلل من خطر تعارض الاعتماديات أو الكتابة فوق ملفات النظام.

/usr/local/bin يحتوي هذا الدليل على الملفات التنفيذية المثبتة في **/usr/local**.

- **Installation_guide:** يمكن أن يكون هذا ملف نصي أو سكريبت يحتوي على تعليمات مفصلة حول كيفية تثبيت البرامج أو تخصيص النظام، إما في البيئة الحية أو بعد التثبيت. قد يتضمن خطوات، نصائح، وأفضل الممارسات.
- **choose-mirror:** هذا السكريبت يساعد على الأرجح في اختيار أقرب أو أسرع مرآة (mirror) لمستودعات حزم Arch Linux ، وهو أمر بالغ الأهمية لتسريع عمليات تنزيل وتحديث الحزم.
- **launch:** اسم عام لسكريبت أو برنامج يُستخدم لبدء تطبيق أو خدمة معينة. يمكن تصميمه لتشغيل بيئة سطح مكتب، أداة معينة، أو عملية تهيئة.
- **livecd-sound:** يشير هذا إلى سكريبت أو برنامج متعلق بإدارة أو تهيئة الصوت في بيئة Live CD/USB. قد يكون مسؤولاً عن تحميل تعريفات الصوت، ضبط مستويات الصوت الأولية، أو تهيئة أجهزة الصوت.

/usr/local/share يحتوي هذا الدليل على ملفات غير قابلة للتنفيذ مرتبطة بالبرامج المثبتة في **/usr/local**.

- **livecd-sound:** قد يحتوي هذا الدليل الفرعي على ملفات تكوين أو بيانات إضافية متعلقة بمعالجة الصوت في البيئة الحية.
- **asound.conf.in:** هذا هو ملف تكوين لـ **ALSA (Advanced Linux Sound Architecture)** اللاحقة **in**. تشير غالبًا إلى أنه ملف قالب سيتم معالجته أو تعديله (ربما بواسطة سكريبت) لإنشاء ملف تكوين **asound.conf** النهائي الذي يستخدمه النظام. يحدد هذا الملف كيفية عمل أجهزة الصوت، المزج، ومكونات الصوت الأخرى.

/usr/share هذا الدليل هو مستودع كبير للملفات المشتركة غير القابلة للتنفيذ، وهي ضرورية لعمل البرامج أو لتوفير واجهة بصرية للمستخدم.

/usr/share/backgrounds يحتوي هذا الدليل على صور خلفية لسطح المكتب يمكن للمستخدمين الاختيار من بينها.

- **bg.png:** هذه صورة خلفية افتراضية أو شائعة الاستخدام.

- **helwan:** هذا دليل فرعي، ومن المحتمل أنه يحتوي على مجموعة من الخلفيات ذات سمة أو تصميم معين يتعلق بـ "حلوان" (اسم يمكن أن يشير إلى مشروع معين، شعار، أو ثقافة).

usr/share/grub يحتوي هذا الدليل على ملفات داعمة لمدير الإقلاع. **GRUB (Grand Unified Bootloader).**

- **themes:** دليل يحتوي على ملفات السمات (themes) لـ GRUB ، مما يسمح بتخصيص شاشة الإقلاع.
- **helwan:** دليل فرعي يمثل سمة مخصصة لـ GRUB ، قد تتضمن شعارات خاصة، صور، وتخطيطات لهذه السمة.
- **usr/share/nemo** يحتوي هذا الدليل على بيانات خاصة بمدير الملفات Nemo ، والذي يُستخدم غالبًا في بيئات سطح المكتب مثل Cinnamon.
- **actions:** هذا الدليل مخصص لـ "إجراءات Nemo المخصصة. (Nemo Actions) "تسمح هذه الإجراءات للمستخدمين بإضافة خيارات جديدة إلى قائمة السياق (right-click context menu) في Nemo ، مما يسهل تنفيذ مهام محددة على الملفات.
- **count_lines.nemo_action:** يحدد هذا الملف إجراء مخصصًا لـ Nemo يقوم بعدّ الأسطر في الملفات المحددة. عندما ينقر المستخدم بزر الماوس الأيمن على ملف (أو مجموعة ملفات) ويختار هذا الإجراء، فإنه سيقوم بتشغيل سكريبت أو أمر يقوم بعدّ الأسطر في تلك الملفات، وربما يعرض النتيجة في نافذة طرفية أو صندوق رسالة. هذا مثال ممتاز على كيفية توسيع وظائف مدير الملفات.
- خاتمة يمثل دليل **usr/** قلب البرامج والتطبيقات في نظام لينكس، حيث يستضيف الغالبية العظمى من البرامج والأدوات والمكتبات التي يستخدمها كل من المستخدمين والمسؤولين. في سياق Archiso ، يعكس هذا الدليل بنية Arch Linux القياسية، مع تسليط الضوء أيضًا على بعض الملفات والسكربتات التي تدعم البيئة الحية وأدوات التخصيص. فهم تنظيم هذا الدليل أمر بالغ الأهمية لفهم كيفية عمل النظام، وكيفية تخصيصه، وكيفية إدارة التطبيقات والخدمات بفعالية.

2.5 Grub

تُعد ملفات تهيئة (GRUB (Grand Unified Bootloader ، مثل grub.cfg و loopback.cfg ، حجر الزاوية لعملية الإقلاع في أنظمة التشغيل الشبيهة بيونكس. يحدد ملف grub.cfg قائمة الإقلاع وخياراتها، بينما يتخصص ملف loopback.cfg في إقلاع الأنظمة مباشرة من صور ISO.

يقدم هذا الفصل شرحًا تفصيليًا لهذين الملفين الحاسمين، ويوضح وظيفة كل أمر ويحلل دوره في عملية الإقلاع. من خلال فهم كيفية تحميل الوحدات (modules) ، وتهيئة وضع الرسومات (graphics mode) ، والتعرف على أنظمة الملفات (file systems) ، يمكن للمستخدمين الحصول على تحكم كامل وتخصيص تجربة الإقلاع الخاصة بهم، سواء لأغراض التصحيح (debugging) ، أو إدارة النظام (system administration) ، أو تثبيت نظام تشغيل جديد.

من خلال هذا الشرح الاحترافي والتقني، ستكتسب فهماً عميقاً للمكونات الداخلية لعملية الإقلاع، مما يتيح لك التعامل مع بيانات GRUB بمرونة وثقة.

grub/ └─ grub/ └─ grub.cfg └─ loopback.cfg

شرح مفصل لملف grub.cfg

ملف grub.cfg هو ملف الإعدادات الرئيسي لـ GRUB (Grand Unified Bootloader) ، وهو محمل إقلاع شائع جدًا في أنظمة التشغيل الشبيهة بيونكس. يحدد هذا الملف كيفية عرض قائمة الإقلاع، وما هي خيارات الإقلاع المتاحة، وكيفية إقلاع كل خيار.

1. تحميل وحدات GRUB

insmod part_gpt

insmod part_msdos

insmod fat

insmod iso9660

insmod ntfs

insmod ntfscomp

insmod exfat

insmod udf

○ insmod module_name: يقوم هذا الأمر بتحميل وحدة (module) محددة في GRUB. الوحدات هي مكونات قابلة للتحميل توسع وظائف GRUB الأساسية.

○ part_gpt و part_msdos: تقوم هذه الأوامر بتحميل الوحدات اللازمة للتعرف على جداول أقسام القرص من نوع GPT (GUID Partition Table) و MBR (Master Boot Record). هذا يسمح لـ GRUB بفهم كيفية تقسيم الأقراص الصلبة.

○ fat, iso9660, ntfs, ntfscomp, exfat, udf: هذه الوحدات ضرورية للتعرف على أنظمة الملفات المختلفة.

▪ fat: نظام الملفات FAT16/FAT32 ، شائع على محركات أقراص USB والأجهزة التخزينية القديمة.

- iso9660: نظام الملفات القياسي المستخدم على أقراص CD/DVD.
- ntfs و ntfscomp: نظام الملفات الخاص بنظام Windows (NTFS) مع دعم الضغط.
- exfat: نظام ملفات شائع للأقراص الكبيرة والتخزين القابل للإزالة.
- udf: نظام الملفات المستخدم غالبًا على أقراص DVD و Blu-ray.

2. إعدادات وضع الرسومات

```
# Use graphics-mode output
```

```
insmod all_video
```

```
insmod font
```

```
if loadfont "${prefix}/fonts/unicode.pf2" ; then
```

```
    set gfxmode="auto"
```

```
    terminal_input console
```

```
    terminal_output console
```

```
fi
```

- insmod all_video: يقوم هذا الأمر بتحميل وحدة تدعم إخراج الفيديو لجميع الأجهزة المتوافقة.
- insmod font: يقوم هذا الأمر بتحميل وحدة الخط (font).
- if loadfont "\${prefix}/fonts/unicode.pf2" ; then ... fi: يتحقق هذا الشرط مما إذا كان يمكن تحميل ملف الخط unicode.pf2 الذي يدعم مجموعة واسعة من الأحرف). إذا نجح التحميل:
- set gfxmode="auto": سيتم تهيئة GRUB لاستخدام أفضل وضع رسومات متاح تلقائيًا.
- terminal_input console و terminal_output console: يوجه هذا الإدخال والإخراج الخاص بـ GRUB إلى وحدة التحكم (console) ، مما يعني أن الرسومات ستظهر مباشرة على شاشة النظام.

3. تمكين وحدة التحكم التسلسلية (Serial Console)

```
# Enable serial console
```

```
if serial --unit=0 --speed=115200; then
```

```
    terminal_input --append serial
```

```
    terminal_output --append serial
```

```
fi
```

○ serial --unit=0 --speed=115200 يقوم هذا الأمر بتهيئة المنفذ التسلسلي للوحدة 0 (عادةً COM1) بسرعة 115200 بت في الثانية.

○ terminal_output --append serial: terminal_input --append serial يضيف هذا المنفذ التسلسلي كوجهة لكل من الإدخال والإخراج. هذا مفيد لتصحيح الأخطاء أو الإدارة عن بعد للنظام عبر اتصال تسلسلي.

4. البحث عن قرص Archiso

```
# Search for the ISO volume
```

```
if [ -z "${ARCHISO_UUID}" ]; then
```

```
if [ -z "${ARCHISO_HINT}" ]; then
```

```
  regexp --set=1:ARCHISO_HINT '^\\([\\^]+)\\)' "${cmdpath}"
```

```
fi
```

```
search --no-floppy --set=root --file '%ARCHISO_SEARCH_FILENAME%' --hint "${ARCHISO_HINT}"
```

```
probe --set ARCHISO_UUID --fs-uuid "${root}"
```

```
fi
```

هذا القسم مخصص للعثور على الوسيط (medium) الذي تم إقلاع GRUB منه، والذي يُفترض أنه يحتوي على ملفات تثبيت Arch Linux (Archiso).

○ if [-z "\${ARCHISO_UUID}"];
يتم التعرف على قرص Archiso بعد.

○ if [-z "\${ARCHISO_HINT}"];
يتم التعرف على قرص Archiso بعد.

○ regexp --set=1:ARCHISO_HINT '^\\([\\^]+)\\)' "\${cmdpath}":
مسار أو اسم ملف من متغير cmdpath، الذي يمثل المسار الذي تم استدعاء GRUB منه.

○ search --no-floppy --set=root --file '%ARCHISO_SEARCH_FILENAME%' --hint "\${ARCHISO_HINT}":
يبحث هذا عن ملف محدد - %ARCHISO_SEARCH_FILENAME% والذي سيتم استبداله بالاسم الفعلي للملف عند الإقلاع) على الأقراص، باستخدام التلميح المقدم إذا كان متاحًا. عند العثور على الملف، يتم تعيين المتغير root ليشير إلى هذا الجهاز.

○ probe --set ARCHISO_UUID --fs-uuid "\${root}":
يحدد هذا الجهاز المحدد بواسطة \${root} للحصول على معرف (Universally Unique Identifier) UUID لنظام الملفات ويخزنه في المتغير ARCHISO_UUID. يتم استخدام هذا المعرف لاحقًا لتحديد قرص Archiso بشكل موثوق.

5. إعدادات قائمة الإقلاع الافتراضية والمهلة الزمنية

Set default menu entry

default=archlinux

timeout=15

timeout_style=menu

○ default=archlinux: يحدد هذا أن إدخال القائمة الأول (الذي له المعرف archlinux) سيكون الإدخال الافتراضي الذي سيتم إقلاعه بعد انتهاء المهلة الزمنية.

○ timeout=15: يضبط هذا وقت الانتظار بالتوازي قبل إقلاع الخيار الافتراضي تلقائيًا.

○ timeout_style=menu: يحدد هذا أن العد التنازلي سيتم عرضه بأسلوب القائمة.

6. تهيئة إقلاع GRUB لإمكانية الوصول (مقتطف الكود):

تهيئة إقلاع GRUB لإمكانية الوصول

play 600 988 1 1319 4

play frequency duration volume: هذا الأمر يشغل نغمة صوتية. قد يُستخدم هذا السطر لإعلام المستخدم بأن GRUB قد بدأ، وهو أمر مفيد للمستخدمين ضعاف البصر. تمثل الأرقام مستويات التردد والمدة ومستوى الصوت.

7. تعريف إدخالات القائمة هنا يتم تعريف الخيارات التي ستظهر للمستخدم عند بدء تشغيل النظام.

7.1. إدخال مثبت

```
Arch Linux: menuentry "Helwan installer(x86_64, UEFI)" --class arch --class gnu-linux --class gnu --class os --id 'archlinux' { set gfxpayload=keep linux /%INSTALL_DIR%/boot/x86_64/vmlinuz-linux archisobasedir=%INSTALL_DIR% archisodevice=UUID=${ARCHISO_UUID} cow_spacesize=5G nouveau.modeset=1 radeon.modeset=1 i915.modeset=1 copytoram=n nvme_load=yes initrd /%INSTALL_DIR%/boot/intel-ucode.img /%INSTALL_DIR%/boot/amd-ucode.img /%INSTALL_DIR%/boot/x86_64/initramfs-linux.img }
```

"Helwan installer(x86_64, UEFI)" --class ... --id 'archlinux': يحدد هذا الأمر تعريف إدخال في القائمة بالاسم، لاحقًا.

set gfxpayload=keep: يحافظ هذا على أي إعدادات رسومية سابقة.

linux /%INSTALL_DIR%/boot/x86_64/vmlinuz-linux ...: هذا هو الأمر الرئيسي لإقلاع نواة لينكس (Linux kernel).

• linux /%INSTALL_DIR%/boot/x86_64/vmlinuz-linux: المسار إلى ملف النواة.

• معلمات النواة: (Kernel Parameters)

○ archisobasedir=%INSTALL_DIR%: يحدد الدليل الأساسي لملفات Archiso.

○ `archisodevice=UUID=${ARCHISO_UUID}`: يحدد جهاز Archiso باستخدام UUID الذي تم العثور عليه مسبقًا.

○ `cow_spacesize=5G`: يضبط حجم مساحة Copy-on-Write (COW) إلى 5 جيجابايت.

○ `nouveau.modeset=1, radeon.modeset=1, i915.modeset=1`: تفعيل إعدادات الأوضاع (modesetting) لبطاقات الرسومات Nvidia (nouveau) ، AMD (radeon) ، و Intel (i915).

○ `copytoram=n`: يمنع نسخ نظام الملفات إلى ذاكرة الوصول العشوائي (RAM).

○ `nvme_load=yes`: يضمن تحميل دعم NVMe لأقراص SSD الحديثة) ميكروًا.

`initrd /%INSTALL_DIR%/boot/intel-ucode.img /%INSTALL_DIR%/boot/amd-ucode.img`
`%INSTALL_DIR%/boot/x86_64/initramfs-linux.img`: يقوم هذا بتحميل نظام ملفات RAM الأولي.

• `intel-ucode.img` و `amd-ucode.img`: ملفات تحديث الميكروكود (microcode) لمعالجات Intel و AMD ، لتحسين الاستقرار والأداء.

• `initramfs-linux.img`: ملف `initramfs` الرئيسي الذي يحتوي على الأدوات اللازمة لبدء تشغيل النظام.

7.2. إدخال خاصية بـ UEFI يظهر هذا القسم فقط إذا كان GRUB يعمل في وضع UEFI.

[`"${grub_platform}" == "efi"]` if تحقق مما إذا كان GRUB يعمل على نظام: [`"${grub_cpu}" == "x86_64"]` if UEFI إذا كان معالج النظام 64 بت ...: `menuentry "Run Memtest86+ (RAM test)"` * :إدخال لتشغيل Memtest86+ ، أداة لاختبار ذاكرة الوصول العشوائي `linux /boot/memtest86+/memtest.efi` * . (RAM) يقوم بتشغيل تطبيق Memtest86+ بتنسيق `EFI. * menuentry` :... "UEFI Shell" إدخال لتشغيل UEFI Shell ، وهي واجهة سطر أوامر يوفرها بيئة `EFI. * insmod chain` :... `chain` ، التي تسمح بـ "سلسلة" تحميل برنامج آخر: `chainloader /shellx64.efi` * . يقوم بتحميل وتشغيل تطبيق UEFI Shell 64 بت [`elif`] : [`"${grub_cpu}" == "i386"]` إذا كان المعالج 32 بت (وهو نادر في أنظمة UEFI الحديثة: `chainloader /shellia32.efi` * : (يقوم بتحميل وتشغيل تطبيق UEFI Shell 32 بت ...: `menuentry 'UEFI Firmware Settings'` :إدخال يسمح للمستخدم بالوصول إلى إعدادات البرامج الثابتة لـ UEFI مباشرة من قائمة `GRUB. * fwsetup` :أمر GRUB للوصول إلى إعدادات UEFI

7.3. إدخال إيقاف التشغيل وإعادة التشغيل ...: `menuentry "System shutdown"` :إدخال لإيقاف تشغيل النظام: `halt` * . أمر GRUB لإيقاف تشغيل النظام ...: `menuentry "System restart"` :إدخال لإعادة تشغيل النظام: `reboot` * . أمر GRUB لإعادة تشغيل النظام.

خاتمة تم تصميم ملف `grub.cfg` هذا ليتم إقلاعه من وسيط (مثل محرك أقراص USB أو DVD) يحتوي على بيئة Archiso. يقوم بتحميل الوحدات النمطية اللازمة، وإعداد واجهة المستخدم (الرسوميات والمتسلسلة)، والبحث عن بيئة Archiso ، ثم يعرض قائمة بخيارات الإقلاع بما في ذلك مثبت Arch Linux ، والأدوات المفيدة مثل Memtest86+ و UEFI Shell ، بالإضافة إلى خيارات إيقاف التشغيل وإعادة التشغيل.

شرح ملف `loopback.cfg`: دليل فني احترافي هذا الملف، `loopback.cfg`، جزء لا يتجزأ من نظام الإقلاع GRUB (Grand Unified Bootloader) وهو مسؤول عن توفير قائمة الخيارات التي تراها عند إقلاع نظام تشغيل من وسيط قابل للإقلاع مثل محرك أقراص USB أو DVD. يركز هذا الملف بشكل خاص على كيفية اكتشاف وتحميل نظام التشغيل من صورة ISO.

آلية وتفصيل الأوامر يهدف `loopback.cfg` إلى تحديد موقع صورة ISO لنظام التشغيل (في هذه الحالة، "Helwan install medium") ثم تهيئة GRUB لإقلاع ذلك النظام. دعنا نفصل الأوامر الرئيسية:

البحث عن وسيط: "\${iso_path}" --no-floppy --set=archiso_img_dev --file "\${iso_path}" يبحث هذا الأمر عن ملف صورة ISO المحدد في المتغير .\${iso_path} الخيار --no-floppy يخبر GRUB بتجاهل محركات الأقراص المرنة، و --set=archiso_img_dev يعين متغيراً داخلياً يسمى archiso_img_dev للإشارة إلى الجهاز الذي تم العثور عليه عليه --probe . "\${archiso_img_dev}" --fs-uuid archiso_img_dev --fs-uuid بعد العثور على الجهاز، يقوم هذا الأمر بفحصه للحصول على معرف نظام الملفات الفريد (UUID) الخاص به ويخزنه في المتغير .archiso_img_dev هذا ضروري لضمان أن GRUB يمكنه دائماً تحديد موقع قسم ISO بشكل موثوق، حتى لو تغير ترتيب الأقراص.

تحديد معرف المنصة (Platform Identifier) الأوامر التي تبدأ بـ ["\${grub_platform}" == ...] if تبني متغير archiso_platform لوصف بيئة الإقلاع الحالية بدقة.

- إذا كان grub_platform هو (UEFI) 'efi' ، فإنه يتحقق من بنية المعالج (grub_cpu) ويضيف 'x64' لـ 'x86_64' أو 'IA32' لـ 'i386' أو ببساطة اسم المعالج إذا كان مختلفاً.

- إذا كان grub_platform هو 'pc' ، يتم تحديد المنصة على أنها 'BIOS'.

- في الحالات الأخرى، يتم استخدام مزيج من grub_cpu و grub_platform لإنشاء معرف فريد. هذا يضمن عرض خيارات الإقلاع الصحيحة بناءً على ما إذا كان النظام يستخدم UEFI أو BIOS ، وبنية المعالج.

إعدادات القائمة الافتراضية

default=archlinux: هذا السطر يحدد أن الخيار الأول في القائمة (الذي يحمل المعرف (archlinux) سيكون هو الخيار الافتراضي، وسيتم إقلاعه تلقائياً بعد انتهاء المهلة.

timeout=15: يحدد هذا السطر وقت الانتظار قبل إقلاع الخيار الافتراضي تلقائياً بـ 15 ثانية.

timeout_style=menu: يحدد هذا السطر أن العد التنازلي للمهلة سيتم عرضه على نمط قائمة.

تحديد خيارات القائمة

كل قسم يبدأ بـ { ... } "..." menuentry يمثل خياراً واحداً في قائمة GRUB.

menuentry "Helwan install medium (%ARCH%, \${archiso_platform})": هذا هو الخيار الرئيسي لتثبيت النظام.

- class arch --class gnu-linux --class gnu --class os: تُستخدم هذه الفئات لتصنيف القائمة وتنسيقها، وغالباً ما تُستخدم في واجهات GRUB الرسومية.

- id 'archlinux': معرف فريد لهذا الخيار، يُستخدم في سطر default=archlinux.

- set gfxpayload=keep: يحافظ على وضع الرسوميات الحالي إذا كان موجوداً.

- linux /%INSTALL_DIR%/boot/%ARCH%/vmlinuz-linux ...: هذا الأمر يحمل نواة نظام التشغيل (vmlinuz-linux).

archisobasedir=%INSTALL_DIR%: يحدد الدليل الأساسي الذي يحتوي على ملفات التثبيت داخل صورة الـ ISO.

img_dev=UUID=\${archiso_img_dev_uuid}: يمرر المعرف الفريد للجهاز الذي يحتوي على صورة الـ ISO إلى النواة.

○ `img_loop="{iso_path}":` يمرر مسار ملف الـ ISO إلى النواة.

- `initrd /%INSTALL_DIR%/boot/%ARCH%/initramfs-linux.img:` هذا يحمل صورة `initramfs` الأولية، الضرورية لبدء تشغيل النظام.

`menuentry "Helwan install medium with speakup screen reader ...":` يشبه هذا الخيار الأول، ولكنه يضيف المعامل `accessibility=on` إلى سطر النواة لتمكين قارئ الشاشة `speakup`، مما يجعله مناسباً للمستخدمين ذوي الاحتياجات الخاصة.

خيارات `Memtest86+`: يتم توفير خيارات لـ `Memtest86+` أداة لاختبار ذاكرة الوصول العشوائي بناءً على ما إذا كان النظام يعمل بنظام `UEFI` أو `BIOS`.

خيار `UEFI Shell:` إذا كان النظام يعمل بنظام `UEFI`، يتم توفير خيار للوصول إلى `UEFI Shell`، والذي يسمح بالتنفيذ المباشر لأوامر `UEFI`.

خيار إعدادات برامج `UEFI` الثابتة: يتيح هذا الخيار للمستخدم الوصول مباشرة إلى إعدادات برامج `UEFI` الثابتة (`BIOS`).

خيارات الإغلاق وإعادة التشغيل `menuentry "System shutdown":` يقوم بإيقاف تشغيل النظام، بينما `menuentry "System restart"` يقوم بإعادة تشغيله.

السياق والتطبيق العملي

يُعد هذا الملف مثالاً عملياً على كيفية تخصيص `GRUB` لبيئات إقلاع معينة، خاصة تلك التي تعتمد على صور الـ `ISO`. يُستخدم ملف `loopback.cfg` في توزيعات `Linux` القائمة على `Arch Linux` مثل `Manjaro` أو `Arch Linux` نفسه عند إنشائه من صورة (`ISO`) لتوفير تجربة مستخدم سلسلة عند الإقلاع من وسائط قابلة للإزالة.

ملخص تقني احترافي

- المرونة: يدعم الملف الإقلاع عبر كل من `UEFI` و `BIOS`، بالإضافة إلى المعالجات 64 بت و 32 بت.
- الكشف التلقائي: يكتشف تلقائياً موقع صورة الـ `ISO` ومعرفها الفريد، مما يجعله مرناً في مواجهة التغييرات في ترتيب الأقراص.
- التخصيص: يسمح بتحديد خيارات متعددة، بما في ذلك وضع إمكانية الوصول والأدوات المساعدة للنظام (`Memtest`)، (`UEFI Shell`).
- قابلية التوسع: يمكن تعديل الخيارات بسهولة لإضافة المزيد من إدخلات القائمة أو تغيير سلوك الإقلاع.

2.6 syslinux/

syslinux/

- ☐ archiso_head.cfg
- ☐ archiso_pxe-linux.cfg
- ☐ archiso_pxe.cfg
- ☐ archiso_sys-linux.cfg
- ☐ archiso_sys.cfg
- ☐ archiso_tail.cfg
- ☐ splash.png
- ☐ syslinux.cfg

مجلدات **syslinux** هي جزء أساسي من عملية إقلاع النظام، خاصةً للأنظمة التي تعتمد على **SYSlinux** أو **ISOLINUX**. يُستخدم هذا النوع من محملات الإقلاع (boot loader) بشكل شائع لإقلاع أنظمة التشغيل من وسائط قابلة للإزالة مثل محركات أقراص **USB** أو عبر الشبكة باستخدام **PXE**.

المجلد الذي قدمته يحتوي على ملفات الإعداد والواجهات الرسومية اللازمة لإقلاع نظام **Arch Linux** من وسائط قابلة للإقلاع. فيما يلي شرح مفصل لكل ملف:

- **syslinux.cfg**: هذا هو ملف إعداد **SYSlinux** الرئيسي. إنه بمثابة نقطة الدخول، حيث يحدد قائمة الإقلاع الافتراضية، والمهلة الزمنية، والخلفية الرسومية، وملفات الإعداد الأخرى التي سيتم تضمينها. يقوم هذا الملف عادةً باستدعاء ملفات إعداد إضافية مثل **archiso_sys.cfg** و **archiso_tail.cfg** لتنظيم الإعدادات.
- **archiso_sys.cfg**: هذا الملف مخصص لإعدادات الإقلاع لأنظمة **Arch Linux** التي تستخدم **BIOS**. يحتوي على إدخالات القائمة التي تسمح للمستخدم بتشغيل مثبت **Arch Linux** أو أدوات أخرى مثل **Memtest86+** في بيئة **BIOS**.
- **archiso_head.cfg**: يحدد هذا الملف بداية ملفات الإعداد. يتضمن أوامر عامة ومقدمة يتم تضمينها في ملفات الإعداد الأخرى، مما يمنع تكرار التعليمات البرمجية.
- **archiso_tail.cfg**: يمثل هذا الملف نهاية ملفات الإعداد. يحتوي على إدخالات القائمة النهائية مثل خيارات الإغلاق وإعادة التشغيل، والتي تظهر دائماً في أسفل قائمة الإقلاع.
- **archiso_pxe.cfg**: هذا الملف مخصص لإقلاع النظام عبر الشبكة باستخدام **PXE** (بيئة التنفيذ قبل الإقلاع). يوفر خيارات الإقلاع للعملاء الذين يتصلون بخادم **PXE** لإقلاع مثبت **Arch Linux** دون الحاجة إلى وسائط مادية.
- **archiso_pxe-linux.cfg**: يُستخدم هذا الملف لتحديد إدخالات القائمة التي تظهر لعملاء **PXE**. يكمل **archiso_pxe.cfg** من خلال توفير التعليمات المحددة لتحميل ملفات النواة و **initrd** عبر الشبكة.
- **archiso_sys-linux.cfg**: هذا الملف مشابه لـ **archiso_pxe-linux.cfg**، ولكنه يُستخدم لإعدادات الإقلاع من وسائط محلية (مثل **USB**) في بيئة **BIOS**.
- **splash.png**: هذه صورة **PNG** تُستخدم كخلفية رسومية لقائمة الإقلاع. توفر واجهة بصرية جذابة بدلاً من الشاشة الافتراضية النصية.

الفكرة وراء فصل ملفات الإعداد هي إنشاء هيكل معياري سهل الإدارة. بدلاً من وجود ملف كبير واحد، يتم تقسيم الإعدادات إلى أجزاء أصغر، لكل منها غرض محدد (مثل الإعدادات المشتركة، أو إعدادات BIOS، أو إعدادات PXE). هذا النهج يسهل التخصيص والصيانة.

syslinux.cfg: ملف إعدادات SYSlinux الرئيسي

يعمل هذا الملف كنقطة دخول أساسية لمُحمّل الإقلاع SYSlinux. يحدد البنية العامة لقائمة الإقلاع ويوجه النظام إلى ملفات الإعدادات الفرعية المناسبة بناءً على بيئة الإقلاع. هذا الهيكل النمطي ضروري للحفاظ على إعدادات منظمة وواضحة.

- **DEFAULT select:** تحدد هذه الأمر أن الخيار الافتراضي الذي سيتم اختياره في قائمة الإقلاع هو **select**.
 - **LABEL select:** يحدد هذا السطر إدخال قائمة الإقلاع المسمى **select**.
 - **COM32 whichsys.c32:** يقوم هذا الأمر بتحميل الوحدة **whichsys.c32**، وهي أداة تم تطويرها خصيصاً لـ **Archiso**. وظيفتها الرئيسية هي فحص بيئة الإقلاع الحالية لتحديد ما إذا كان الإقلاع من وسائط محلية أو عبر الشبكة (PXE).
 - **sys-iso- sys-pxe-pxe-APPEND:** يمرر هذا الأمر معاملات إلى الوحدة **whichsys.c32**. تُستخدم هذه المعاملات لتوجيه الوحدة إلى "منصات" محددة.
 - **pxe-pxe-pxe:** إذا تم اكتشاف إقلاع PXE، فسيتم إعادة التوجيه إلى تسمية **pxe**.
 - **sys-sys-sys:** إذا تم اكتشاف الإقلاع من وسائط محلية (مثل محرك أقراص USB)، فسيتم إعادة التوجيه إلى تسمية **sys**.
 - **iso-sys-iso:** يضمن هذا المعامل أن الإقلاع من صورة ISO يذهب أيضاً إلى تسمية **sys**.
 - **LABEL pxe:** تحدد هذه التسمية نقطة البداية لإعدادات إقلاع PXE.
 - **CONFIG archiso_pxe.cfg:** يوجه هذا الأمر SYSlinux لتحميل ملف الإعدادات **archiso_pxe.cfg**، الذي يحتوي على جميع الخيارات والإعدادات للإقلاع عبر الشبكة.
 - **LABEL sys:** تحدد هذه التسمية نقطة البداية لإعدادات الإقلاع من الوسائط المحلية.
 - **CONFIG archiso_sys.cfg:** يوجه هذا الأمر SYSlinux لتحميل ملف الإعدادات **archiso_sys.cfg**، الذي يحتوي على خيارات الإقلاع الخاصة بالإقلاع من محرك أقراص USB أو DVD.
- باختصار، يعمل ملف **syslinux.cfg** كلوحة تحكم ذكية، تستخدم **whichsys.c32** لتحديد بيئة التشغيل تلقائياً، ثم تمرير التحكم إلى ملف الإعدادات الصحيح، مما يوفر تجربة إقلاع سلسة وفعالة للمستخدم.

archiso_sys.cfg: إعدادات الإقلاع لـ BIOS

يعمل هذا الملف كمركز رئيسي لإعدادات الإقلاع لأنظمة Arch Linux التي يتم إقلاعها من وسائط محلية في بيئة BIOS (نظام الإدخال/الإخراج الأساسي). دوره الأساسي هو استدعاء ملفات إعدادات أخرى بطريقة منظمة لإنشاء قائمة إقلاع كاملة ومتكاملة.

- **INCLUDE archiso_head.cfg:** يوجه هذا الأمر SYSlinux لتضمين محتوى ملف archiso_head.cfg. كما ذكرنا سابقًا، يحتوي هذا الملف على إعدادات عامة وتمهيدية تُستخدم عبر جميع ملفات الإعدادات. تضمن هذه الممارسة الاتساق وتمنع تكرار التعليمات البرمجية.
 - **DEFAULT arch64:** يحدد هذا الأمر أن الخيار الافتراضي الذي سيتم اختياره في قائمة الإقلاع هو arch64.
 - **TIMEOUT 150:** يحدد هذا السطر وقت الانتظار بالسنت ثانية (100/1 من الثانية) قبل إقلاع الخيار الافتراضي تلقائيًا. القيمة 150 تعادل 1.5 ثانية، مما يمنح المستخدم وقتًا كافيًا لاختيار خيار مختلف.
 - **INCLUDE archiso_sys-linux.cfg:** هذا أمر أساسي يوجه SYSlinux لتضمين محتوى ملف archiso_sys-linux.cfg. يحتوي هذا الملف على إدخالات القائمة الفعلية، مثل خيار تثبيت Arch Linux.
 - **INCLUDE archiso_tail.cfg:** يمثل هذا الأمر الخطوة النهائية في بناء قائمة الإقلاع، حيث يتضمن محتوى ملف archiso_tail.cfg، الذي يحتوي على الخيارات النهائية مثل الإغلاق وإعادة التشغيل.
- باختصار، يعمل archiso_sys.cfg كمنسق يجمع أجزاء مختلفة من الإعدادات لإنشاء قائمة إقلاع كاملة مصممة خصيصًا لبيئات BIOS، مما يضمن توفر جميع الخيارات الضرورية للمستخدم بتسلسل منطقي.

archiso_head.cfg: رأس الإعدادات المشترك

يعمل هذا الملف كرأس مشترك لملفات إعدادات SYSLinux المختلفة. الغرض الأساسي منه هو تحديد الإعدادات العامة والعناصر المرئية التي يتم مشاركتها عبر بيئات الإقلاع المختلفة (مثل الوسائط المحلية، PXE). من خلال تركيز هذه الإعدادات، فإنه يضمن الاتساق ويتجنب تكرار التعليمات البرمجية، مما يلتزم بأفضل الممارسات في التصميم النمطي.

يمكن تقسيم محتوى الملف إلى ثلاثة أقسام رئيسية: إعداد وحدة التحكم والواجهة (UI)، أبعاد وتخطيط القائمة، ومخطط الألوان.

إعداد وحدة التحكم والواجهة (Console and UI Setup)

- **SERIAL 0 115200**: يقوم هذا الأمر بتكوين وحدة تحكم تسلسلية. يقوم بتنشيط المنفذ التسلسلي 0 (عادةً COM1) ويضبط سرعة الاتصال على 115,200 بود. هذا ضروري للإدارة عن بعد للنظام والتصحيح، حيث يسمح للمسؤولين بالتفاعل مع مُحمل الإقلاع من طرف بعيد.
- **UI vesamenu.c32**: يقوم بتحميل الوحدة VesaMenu، وهي واجهة مستخدم رسومية (GUI) لـ SYSLinux. يستبدل القائمة النصية الافتراضية بواجهة أكثر جاذبية بصرياً، مما يتيح استخدام الخلفيات وتخصيص القائمة المتقدم.
- **MENU TITLE Helwan**: يحدد هذا العنوان الذي يظهر في أعلى قائمة الإقلاع، وفي هذه الحالة، "Helwan".
- **MENU BACKGROUND splash.png**: يحدد صورة الخلفية لقائمة الإقلاع، ويشير إلى ملف splash.png. يوفر هذا مظهرًا احترافيًا ومميزًا لمُحمل الإقلاع.

أبعاد وتخطيط القائمة (Menu Dimensions and Layout)

يحدد هذا القسم التخطيط الدقيق لقائمة الإقلاع على الشاشة. القيم بوحدة الأحرف، مما يسمح بالتحكم الدقيق في مظهر القائمة.

- **MENU WIDTH 78**: يضبط عرض نافذة القائمة الرئيسية على 78 حرفاً.
- **MENU MARGIN 4**: يضبط الهامش الأيسر للقائمة على 4 أحرف.
- **MENU ROWS 7**: يحدد عدد صفوف القائمة المرئية.
- **MENU VSHIFT 10**: يقوم بإزاحة القائمة رأسياً بمقدار 10 صفوف.
- **MENU TABMSGROW 14, MENU CMDLINEROW 14, MENU HELPMMSGROW 16, MENU HELPMMSGENDROW 29**: تحدد هذه الأوامر المواضع الرأسية لمنطقة رسالة علامة التبويب (tab message)، سطر الأوامر (command line)، ومنطقة رسالة المساعدة (help message)، مما يضمن عدم تداخلها مع عناصر القائمة الرئيسية.

مخطط الألوان (Color Scheme)

هذا قسم حاسم للتخصيص المرئي. يحدد لون كل عنصر في قائمة الإقلاع باستخدام تنسيق لون ست عشري (hexadecimal).

- **MENU COLOR border 30;44 #40ffffff #a0000000 std**: يضبط لون حدود القائمة. تمثل القيم لون المقدمة، لون الخلفية، ورموز ست عشرية محددة للشفافية وخط الألوان.
- تتبع أسطر **MENU COLOR** اللاحقة (مثل title, sel, unsel) نفس التنسيق، مما يسمح بالتحكم التفصيلي في ألوان العنوان، العناصر المحددة، العناصر غير المحددة، نص المساعدة، ومكونات القائمة الأخرى.

سلوك القائمة (Menu Behavior)

- **MENU CLEAR**: يقوم هذا الأمر بمسح الشاشة قبل عرض القائمة، مما يضمن عرضاً نظيفاً.
 - **MENU IMMEDIATE**: يتسبب هذا الأمر في عرض القائمة فوراً دون انتظار ضغط مفتاح من المستخدم.
- باختصار، ملف `archiso_head.cfg` ليس ملفاً قابلاً للإقلاع بحد ذاته. إنه مكون أساسي يتم تضمينه بواسطة ملفات الإعدادات الأخرى لإنشاء بيئة إقلاع متسقة ومصممة جيداً وعملية للغاية. دوره هو تحديد "الشكل والمظهر" والسلوك الأساسي لقائمة `SYSlinux`، وفصل العرض عن الوظيفة.
- `archiso_tail.cfg`: قوائم المرافق الشائعة يحتوي هذا الملف على المجموعة النهائية من خيارات المرافق وإدارة النظام التي يتم عادةً إلحاقها بنهاية قائمة الإقلاع. توفر هذه الإدخالات وظائف أساسية للمستخدمين الذين يتعاملون مع وسائط الإقلاع، مما يوفر طرقاً لإقلاع الأنظمة المثبتة، وإجراء تشخيصات الأجهزة، وإدارة حالة طاقة النظام.
- **LABEL existing**
 - الغرض: يسمح هذا الإدخال للمستخدم بإقلاع نظام تشغيل موجود بالفعل على القرص الصلب للكمبيوتر. هذا أمر بالغ الأهمية للبيئات الحية التي تعمل أيضاً كمدير إقلاع للأنظمة المثبتة.
 - **TEXT HELP ... ENDTEXT**: توفر هذه الكتلة معلومات مفيدة للمستخدم. توضح أن هذا الخيار يقوم بإقلاع نظام تشغيل مثبت وتوجه المستخدم للضغط على **TAB** لتعديل معلمات الإقلاع (أرقام القرص والقسم).
 - **MENU LABEL Boot existing OS**: يضبط النص المرئي لعنصر القائمة هذا.
 - **COM32 chain.c32**: يقوم بتحميل وحدة `chain.c32`، المسؤولة عن تحميل مُحَمِّل إقلاع آخر (عادةً ما يكون من قطاع الإقلاع لنظام التشغيل المثبت).
 - **APPEND hd0 0**: تخبر هذه المعلمة `chain.c32` بالإقلاع من القرص الصلب الأول (`hd0`) والقسم الأول (`0`) على هذا القرص. هذه طريقة شائعة لتحديد قسم الإقلاع الأساسي للنظام المثبت.

LABEL memtest

- الغرض: يبدأ هذا الإدخال **+Memtest86**، وهي أداة شائعة الاستخدام لاختبار ذاكرة الوصول العشوائي (**RAM**) للنظام. يعد إجراء اختبار **RAM** خطوة تشخيصية أساسية لاستبعاد أخطاء الذاكرة كسبب لعدم استقرار النظام أو مشكلات الأداء.
- **(RAM test) +Memtest86 (MENU LABEL Run Memtest86)**: يضبط النص المرئي لعنصر القائمة هذا، مما يشير بوضوح إلى وظيفته.
- **LINUX /boot/memtest86+/memtest**: يقوم هذا الأمر بتحميل الملف التنفيذي لـ **+Memtest86**، والذي يقع داخل دليل `+boot/memtest86/` على وسائط الإقلاع.

• LABEL hdt

- الغرض: يقوم هذا الإدخال بتشغيل HDT (أداة الكشف عن الأجهزة)، وهي أداة مساعدة توفر معلومات مفصلة حول مكونات أجهزة الكمبيوتر. هذا لا يقدر بثمن لاستكشاف مشكلات توافق الأجهزة أو ببساطة فهم تكوين النظام.
- (MENU LABEL Hardware Information (HDT: يضبط النص المرئي لعنصر القائمة هذا.
- COM32 hdt.c32: يقوم بتحميل وحدة HDT.
- APPEND modules_alias=hdt/modalias.gz pciids=hdt/pciids.gz: تمرر هذه المعلومات ملفات التكوين إلى HDT. يساعد HDT modules_alias.gz على ربط أجهزة الأجهزة بوحدات النواة الصحيحة، بينما يوفر pciids.gz خريطة لمعرفة أجهزة PCI لتحديد أفضل.

• LABEL reboot

- الغرض: يسمح هذا الخيار للمستخدم بإعادة تشغيل الكمبيوتر مباشرة من قائمة الإقلاع.
- TEXT HELP ... ENDTEXT: يحدد نص المساعدة أن برنامج الكمبيوتر الثابت يجب أن يدعم APM (إدارة الطاقة المتقدمة) لكي يعمل هذا الخيار بشكل صحيح.
- MENU LABEL Reboot: يضبط النص المرئي لعنصر القائمة هذا.
- COM32 reboot.c32: يقوم بتحميل وحدة reboot.c32، التي تتعامل مع عملية إعادة تشغيل النظام.

• LABEL poweroff

- الغرض: يسمح هذا الخيار للمستخدم بإيقاف تشغيل الكمبيوتر مباشرة من قائمة الإقلاع.
- TEXT HELP ... ENDTEXT: على غرار reboot، يشير نص المساعدة هذا إلى أن برنامج الكمبيوتر الثابت يجب أن يدعم APM (إدارة الطاقة المتقدمة) لوظيفة إيقاف التشغيل.
- MENU LABEL Power Off: يضبط النص المرئي لعنصر القائمة هذا.
- COM32 poweroff.c32: يقوم بتحميل وحدة poweroff.c32، التي تغلق النظام وتطفئ الأجهزة بشكل سليم.

يشكل أساسي، يقوم archiso_tail.cfg بدمج أدوات النظام الأساسية في قائمة سهلة الاستخدام، مما يكمل خيارات تثبيت نظام التشغيل الأساسي أو البنية الحية. غالبًا ما تكون هذه هي العناصر الأخيرة التي يتم تقديمها، مما يوفر قدرات احتياطية وتشخيصية.

archiso_pxe.cfg: تكوين الإقلاع عبر الشبكة (PXE) يعد هذا الملف جزءاً لا يتجزأ من عملية إعداد الإقلاع باستخدام بروتوكول PXE (بيئة التنفيذ المسبق للإقلاع). هدفه الأساسي هو توفير خيارات الإقلاع الضرورية لجهاز العميل الذي يطلب الإقلاع عبر الشبكة، مما يمكنه من تحميل نظام تشغيل Arch Linux دون الحاجة إلى وسائط تخزين محلية (مثل USB أو DVD).

- **INCLUDE archiso_head.cfg**: يتضمن هذا الأمر ملف **archiso_head.cfg**. كما تم شرحه سابقاً، يوفر هذا الملف إعدادات مشتركة، مثل إعدادات الواجهة الرسومية (VesaMenu)، والعنوان، وصورة الخلفية (**splash.png**). يضمن هذا التضمين أن جميع الأجهزة التي يتم إقلاعها عبر PXE تشترك في نفس المظهر الأساسي والإعداد الأولي.
- **INCLUDE archiso_pxe-linux.cfg**: هذا هو الأمر المحوري داخل هذا الملف. يتضمن ملف **archiso_pxe-linux.cfg**، الذي يحتوي على إدخال القائمة الفعلية للإقلاع عبر PXE. هنا، يتم تحديد المسارات إلى نواة Linux وصورة **initramfs** التي سيتم تحميلها عبر الشبكة، جنباً إلى جنب مع أي معلمات نواة ضرورية للإقلاع عبر الشبكة.
- **INCLUDE archiso_tail.cfg**: يكمل هذا الأمر البناء النهائي لقائمة الإقلاع عن طريق تضمين ملف **archiso_tail.cfg**. يحتوي هذا الملف عادةً على خيارات الإقلاع النهائية مثل "إعادة التشغيل" و "إيقاف التشغيل"، والتي تظهر في نهاية القائمة بغض النظر عن طريقة الإقلاع (PXE أو محلي).

باختصار، يعمل ملف **archiso_pxe.cfg** كمنسق لعملية الإقلاع عبر الشبكة. فهو يطبق أولاً الإعدادات الرسومية والبيئية الشائعة من **archiso_head.cfg**، ثم يقوم بتحميل الخيارات المحددة للإقلاع عبر PXE من **archiso_pxe-linux.cfg**، وأخيراً يلحق خيارات الإغلاق القياسية من **archiso_tail.cfg**. يضمن هذا الهيكل المعياري تجربة إقلاع متسقة وفعالة عبر الشبكة.

خيارات قائمة الإقلاع عبر الشبكة في **archiso_pxe-linux.cfg** يحتوي هذا الملف على إدخال القائمة المحددة التي تظهر عند إقلاع نظام عبر الشبكة. يحدد كيفية تحميل نواة Linux وملفات النظام الأولية (**initramfs**) من خادم شبكة باستخدام بروتوكولات مختلفة.

• LABEL arch64_nbd

- **TEXT HELP** و **ENDTEXT**: يوفر نص مساعدة يشرح أن هذا الخيار يسمح بإقلاع وسائط تثبيت حلوان عبر NBD (جهاز كتلة الشبكة)، مما يتيح تثبيت النظام أو صيانتة.
- **(MENU LABEL Helwan install medium (x86_64, NBD**: يحدد النص الذي سيتم عرضه في قائمة الإقلاع.
- **LINUX ::/%INSTALL_DIR%/boot/x86_64/vmlinuz-linux**: يحدد المسار إلى نواة Linux.
- **INITRD ::/%INSTALL_DIR%/boot/intel-ucode.img,::/%INSTALL_DIR%/boot/amd-ucode.img,::/%INSTALL_DIR%/boot/x86_64/initramfs-linux.img**: يحدد المسارات إلى ملفات **initrd**. تشير البادئة :: إلى أن هذه الملفات يجب أن يتم تحميلها عبر الشبكة.
- **APPEND archisobasedir=%INSTALL_DIR% archisodevice=UUID=%ARCHISO_UUID% archiso_nbd_srv=\${pxeserver} cms_verify=y**: يمرر معلمات النواة الضرورية لإعداد بيئة التثبيت، بما في ذلك خادم NBD.

• LABEL arch64_nfs

- TEXT HELP و ENDTEXT: يوفر نص مساعدة يشرح أن هذا الخيار يسمح بإقلاع وسائط تثبيت حلوان عبر NFS (نظام ملفات الشبكة)، مما يتيح تثبيت النظام أو صيانتة.
- (MENU LABEL Helwan install medium (x86_64, NFS): يحدد النص الذي سيتم عرضه في قائمة الإقلاع.
- LINUX ::/%INSTALL_DIR%/boot/x86_64/vmlinuz-linux: يحدد المسار إلى نواة Linux.
- INITRD ::/%INSTALL_DIR%/boot/intel-ucode.img,::/%INSTALL_DIR%/boot/amd-ucode.img,::/%INSTALL_DIR%/boot/x86_64/initramfs-linux.img: يحدد المسارات إلى ملفات .initrd
- APPEND archisobasedir=%INSTALL_DIR% archi-so_nfs_srv=\${pxeserver}:/run/archiso/bootmnt cms_verify=y: يمرر معلومات النواة لتحديد خادم NFS ومسار التثبيت.

• LABEL arch64_http

- TEXT HELP و ENDTEXT: يوفر نص مساعدة يشرح أن هذا الخيار يسمح بإقلاع وسائط تثبيت حلوان عبر HTTP، مما يتيح تثبيت النظام أو صيانتة.
- (MENU LABEL Helwan install medium (x86_64, HTTP): يحدد النص الذي سيتم عرضه في قائمة الإقلاع.
- LINUX ::/%INSTALL_DIR%/boot/x86_64/vmlinuz-linux: يحدد المسار إلى نواة Linux.
- INITRD ::/%INSTALL_DIR%/boot/intel-ucode.img,::/%INSTALL_DIR%/boot/amd-ucode.img,::/%INSTALL_DIR%/boot/x86_64/initramfs-linux.img: يحدد المسارات إلى ملفات .initrd
- APPEND archisobasedir=%INSTALL_DIR% archi-so_http_srv=http://\${pxeserver}/: يمرر معلومات النواة لتحديد خادم HTTP.

فهم archiso_sys-linux.cfg للإقلاع عبر BIOS

يُعرّف ملف التكوين هذا، archiso_sys-linux.cfg، خيارات قائمة الإقلاع للأنظمة التي تقف عبر BIOS في بيئة إقلاع شبكية، وتحديدًا لوسائط تثبيت "Helwan". يوضح هذا الملف كيفية تحميل نواة Linux وقرص RAM الأولي (initramfs) لبدء عملية التثبيت أو الصيانة.

فيما يلي تفصيل لكل مكون:

- **arch64: LABEL** هذا هو مُعرّف فريد لـ entry الإقلاع المحدد هذا. بهذه الطريقة يشير مُحمّل الإقلاع إلى هذا التكوين المحدد.
- **TEXT HELP ... ENDTEXT:** توفر هذه الكتلة نص مساعدة وصفي يظهر عند اختيار هذا الخيار في القائمة. يُعلم المستخدم أن هذا الخيار مخصص لإقلاع وسيط تثبيت Helwan على نظام BIOS ويمكن استخدامه لتثبيت Helwan أو إجراء صيانة للنظام.
- **MENU LABEL Helwan install medium (x86_64, BIOS):** هذا هو النص سهل الاستخدام الذي سيتم عرضه في قائمة الإقلاع، مما يسمح للمستخدمين بتحديد هذا الخيار واختياره بسهولة.
- **LINUX (%INSTALL_DIR%/boot/x86_64/vmlinuz-linux):** يحدد هذا التوجيه موقع نواة Linux (vmlinuz-linux). (linux) يشير الجزء /%INSTALL_DIR%/ إلى أن هذه النواة هي جزء من وسائط التثبيت، وهي موجودة داخل الدليل boot/x86_64/.
- **INITRD (%INSTALL_DIR%/boot/intel-ucode.img,%/INSTALL_DIR%/boot/amd-ucode.img,%/INSTALL_DIR%/boot/x86_64/initramfs-linux.img):** يحدد هذا التوجيه صور قرص RAM الأولي التي يتم تحميلها مع النواة.
 - **intel-ucode.img و amd-ucode.img** هي تحديثات دقيقة للمعالجات من Intel و AMD على التوالي. تساعد هذه في ضمان التهيئة الصحيحة للأجهزة واستقرارها.
 - **initramfs-linux.img** هو نظام ملفات RAM الأولي الرئيسي، والذي يحتوي على التعريفات والأدوات الأساسية اللازمة لتحميل نظام الملفات الجذر والاستمرار في عملية الإقلاع.
- **APPEND archisobasedir=%INSTALL_DIR% archisodevice=UUID=%ARCHISO_UUID% cow_spacesize=5G nouveau.modeset=1 radeon.modeset=1 i915.modeset=1 copytoram=n nvme_load=yes:** هذا السطر معلمات نواة مختلفة (أو وسائط إقلاع) تقوم بتكوين البيئة للتثبيت.
 - **archisobasedir=%INSTALL_DIR%:** يخبر النظام بمكان وجود ملفات Archiso التثبيت الرئيسية.
 - **archisodevice=UUID=%ARCHISO_UUID%:** يحدد الجهاز الذي يحتوي على وسائط التثبيت باستخدام مُعرّفه الفريد عالميًا (UUID). هذا يضمن أن النظام يمكنه العثور على مصدر التثبيت بشكل موثوق.
 - **cow_spacesize=5G:** يضبط حجم مساحة النسخ عند الكتابة (Copy-on-Write - CoW) إلى 5 جيجابايت. يُستخدم هذا للتغييرات المؤقتة أثناء البيئة الحية.
 - **nouveau.modeset=1, radeon.modeset=1, i915.modeset=1:** تمكّن هذه المعلمات الإعداد الخاص بالوضع (kernel mode setting) لمُحركات الرسومات Nvidia (nouveau) و AMD (radeon) و Intel (i915) على التوالي. يساعد هذا في الحصول على إخراج رسومات مناسب في وقت مبكر من عملية الإقلاع.

○ **copytoram=n:** يشير إلى أنه لا ينبغي نسخ وسائط التثبيت بالكامل إلى ذاكرة الوصول العشوائي (RAM) هذا يوفر الذاكرة ولكنه يعني أن النظام سيصل إلى ملفات التثبيت من موقعها الأصلي (على سبيل المثال، الشبكة أو USB).

○ **nvme_load=yes:** يتضمن تحميل تعريفات وحدات تخزين NVMe ، وهو أمر بالغ الأهمية للأنظمة التي تستخدم أقراص NVMe SSD.

يشكل أساسي، يوفر entry الموجود في archiso_sys-linux.cfg هذا تكوينًا محددًا لإقلاع بيئة تثبيت Helwan على أنظمة BIOS الأقدم، مما يضمن تهيئة جميع وحدات النواة والمعلومات الضرورية بشكل صحيح لعملية إقلاع وتثبيت ناجحة.

2.6 أسرار الدليل /etc/

مقدمة

عندما نتحدث عن بناء توزيع لينكس باستخدام **archiso**، غالبًا ما يركز المطورون على اختيار الحزم، أو تكوين بيئة سطح المكتب، أو إضافة بعض نصوص المساعدة البرمجية. ومع ذلك، الحقيقة هي أن كل هذه التفاصيل، على الرغم من أهميتها، لا تخدم السطح فقط. الجوهر، الشخصية الحقيقية للتوزيع، تُزرع في مكان واحد: دليل **/etc/**.

هذا الدليل ليس مجرد "صندوق إعدادات" كما قد يتخيل البعض؛ إنه عقل النظام. إنه يحتوي على كل ما يحدد هوية التوزيع:

- كيف يتعامل النظام مع المستخدمين وصلاحياتهم.
 - كيف تتم إدارة الشبكات والخدمات.
 - لغة التوزيع، والمنطقة الزمنية، واسم الشبكة.
 - كيف يقلع النظام، وأي نواة وبرامج تشغيل يتم تحميلها.
 - المستودعات التي يثق بها مدير الحزم، **pacman**.
 - كيف يقوم المستخدم بتسجيل الدخول لأول مرة إلى بيئة سطح المكتب، وما هي الإعدادات الافتراضية التي يحصل عليها.
- كل هذه التفاصيل وأكثر مخزنة في **/etc/** هنا يكمن خطر هذا الدليل: خطأ صغير بداخله يمكن أن يمنع النظام من الإقلاع، ولمسة دقيقة يمكن أن تغير تجربة المستخدم بالكامل.

من يونس إلى حلوان لينكس: قصة دليل صغير

تاريخيًا، بدأ اسم **/etc/** اختصارًا لـ **"et cetera"** (إلخ)، وهو مكان لتجميع الملفات التي ليس لها مكان أفضل. ولكن مع تطور يونس ثم جنو/لينكس، تحول تدريجيًا إلى مركز ثقل النظام، حتى أصبح اليوم بمثابة "دستور" التوزيع.

من خلال فحص **/etc/** بعناية، يمكن فهم فلسفة أي نظام تشغيل: هل هو موجه للمطورين؟ للمبتدئين؟ للخوادم؟ أو لمجتمع معين، مثل هوية حلوان لينكس؟

الخريطة **/etc/**: في حلوان لينكس

لتسهيل الفهم، يمكننا اعتبار **/etc/** بمثابة شجرة ذات جذور متفرعة. في توزيع حلوان لينكس، يأخذ هذا الدليل هيكلًا غنيًا، كما هو موضح أدناه:

etc/

- إدارة المستخدمين (passwd, shadow, group, gshadow)
- هوية النظام (hostname, locale.conf, localtime)
- الإقلاع والتشغيل (grub, mkinitcpio*, modprobe.d)
- مدير الحزم pacman (pacman.conf, pacman.d/hooks)
- النظام الرسومي (lightdm, skel, X11)
- الشبكات (resolv.conf, systemd/network, ssh)
- الأمان والصلاحيات (sudoers.d, polkit, motd)
- الخدمات systemd (network, display, cloud-init, zram ...)

ل — إعدادات المستخدم الرسومية (xdg/reflector, cinnamon configs) هذه ليست مجرد أسماء ملفات؛ كل عنصر هو قصة منفصلة. على سبيل المثال:

- يمثل passwd و shadow العمود الفقري لإدارة الحسابات.
- يحدد pacman.conf بدقة من أين يأتي البرنامج وكيف يتم التحقق من سلامته.
- يقوم lightdm.conf و slick-greeter.conf بتكوين الشاشة الأولى التي يراها المستخدم.
- تضمن خدمة reflector.service تحديث المرايا تلقائيًا للحصول على تثبيتات وتحديثات سريعة ومستقرة.
- يعكس الدليل skel/Templates/hel-files البصمة الفريدة لحلوان لينكس، حيث يقدم ملفات جاهزة تسهل عمل المبرمجين ومنشئي المحتوى.

بيئة حية مقابل بيئة مثبتة: شخصيتان في كيان واحد

أحد أذكى التفاصيل التي تظهر داخل /etc/ هو التباين بين البيئة الحية (Live) والبيئة المثبتة (Installed)

- في البيئة الحية: إعدادات مؤقتة، تسجيل دخول تلقائي، صلاحيات مفتوحة، وخدمات موجهة للتجربة.
- في البيئة المثبتة: مستخدم جديد بكلمة مرور، صلاحيات مشددة، خدمات تجريبية متوقفة، ونقل النظام إلى بيئة آمنة وكاملة.

هذا التباين ليس من قبيل الصدفة؛ إنه يُدار بعناية من خلال الملفات والخدمات داخل /etc/.

هدف هذا الفصل

في هذا الفصل، لن نراجع الملفات فحسب؛ بل سنتعمق في أعماقها. سنشرح ما يفعله كل ملف، وكيف يتفاعل مع بقية النظام، ولماذا هو مهم في سياق بناء توزيع. سنقسم الرحلة إلى ثمانية محاور رئيسية:

- إدارة المستخدم والأمان.

- هوية النظام.

- الإقلاع والتشغيل.

- مدير الحزم. pacman.

- النظام الرسومي وواجهة تسجيل الدخول.

- الشبكات والاتصالات.

- الأمان والصلاحيات.

- systemd والخدمات.

- إعدادات المستخدم الرسومية.

سنركز على أكثر من ستين ملفًا ودليلاً، بعضها أساسي لعمل النظام، بينما يضيف البعض الآخر لمسة خاصة لهوية حلوان لينكس.

خاتمة المقدمة

رحلتنا إلى `/etc/` ليست مجرد درس في ملفات نصية؛ إنها رحلة لاكتشاف شخصية التوزيعة من الداخل. أي شخص يقرأ هذا الدليل بعمق سيدرك أن كل ملف، مهما بدا بسيطاً، هو خيط في نسيج كامل.

الآن، لنبدأ من البداية: إدارة المستخدم والأمان، حيث تُكتب القواعد الأولى للعبة.

2.6.2 إدارة المستخدم والأمان

ملف `/etc/passwd`

يُعد هذا الملف العمود الفقري لتعريف المستخدمين في أي نظام Unix/Linux. يمثل كل سطر مستخدماً، وتفصل الحقول بداخله بواسطة ::.

- `root:x:0:0:root:/root:/usr/bin/zsh`

- `liveuser:x:1000:1000::/home/liveuser:/bin/bash`

- `root:` المستخدم ذو أعلى الامتيازات (المشرف الخارق).

- `UID = 0:` صلاحيات مطلقة.

- `shell = zsh` بدلاً من `bash`: بقرار تصميمي في Helwan Linux لتجربة أكثر حداثة.

- `liveuser:` المستخدم لجلسة Live.

- `UID = 1000` أول مستخدم عادي.

- `shell = bash` بسيط وسهل للمبتدئين.

الفلسفة: في Unix ، "كل شيء ملف". حتى تعريفات المستخدم هي مجرد نصوص يمكن قراءتها وتعديلها.

ملف `/etc/shadow`

يخزن هذا الملف كلمات المرور (بشكل مشفر).

- `root::14871::::::`

- `liveuser::14871::::::`

الحقول الفارغة تعني أن الحسابات ليس لها كلمات مرور. النتيجة: تسجيل دخول مباشر في بيئة Live دون الحاجة إلى كلمة مرور.

ملف `/etc/group`

تحدد المجموعات الأذونات المشتركة.

- `wheel:x:998:liveuser`

- `network:x:90:liveuser`

- `audio:x:995:liveuser`

- `video:x:986:liveuser`

- `storage:x:988:liveuser`

- `wheel:` للوصول إلى `sudo`.

- `network:` لإدارة الشبكة.

- `audio/video/storage:` لتشغيل الوسائط والتخزين.

النتيجة: `liveuser` جاهز فورًا لتجربة كاملة دون قيود.

ملف `/etc/gshadow`

يحمل هذا الملف نفس فكرة `/etc/group` ولكن للمصادقة.

- `wheel:!!!:liveuser`

- `!!!` يشير إلى عدم تعيين كلمة مرور للمجموعة.

ملف `/etc/sudoers.d/g_wheel`

`%wheel`

`ALL=(ALL) NOPASSWD: ALL`

يمكن لأي عضو في مجموعة `wheel` (مثل `liveuser`) تنفيذ أوامر `sudo` دون كلمة مرور. هذا مفيد لبيئة `Live` ولكنه عادةً ما يتم تغييره بعد التنصيب لتأمين النظام.

2.6.3. إعدادات النظام العامة

`etc/hostname/`

`archiso`

يحدد هذا الملف اسم المضيف الافتراضي للنظام. في هذه الحالة، `archiso` هو اسم بسيط وواضح يشير إلى أن النظام مبني على `ArchISO`. يمكن تغييره بسهولة أثناء عملية التنصيب.

`etc/locale.conf/`

`LANG=C.UTF-8`

تحدد هذه التهيئة إعدادات اللغة للنظام. `C.UTF-8` هو خيار محايد ومتوافق دوليًا يمنع بشكل فعال مشاكل ترميز الأحرف. يضمن هذا إخراجًا متسقًا من البرامج والسجلات عبر النظام.

`etc/localtime/`

`usr/share/zoneinfo/UTC/`

يشير هذا الإدخال إلى المنطقة الزمنية للنظام. افتراضيًا، يتم تعيينه إلى التوقيت العالمي المنسق (`UTC`). هذا يبسط التنصيب العالمي ويسمح بتخصيص المنطقة الزمنية في مرحلة لاحقة.

2.6.4. إدارة الإقلاع والتهيئة

etc/default/grub/

يوفر هذا الملف تحكماً كاملاً في محمل الإقلاع GRUB.

- GRUB_DISTRIBUTOR="Helwan": يحدد الهوية المرئية للتوزيعة داخل قائمة GRUB.

- GRUB_CMDLINE_LINUX_DEFAULT="loglevel=7 audit=0":

- loglevel=7: يعرض رسائل مفصلة أثناء الإقلاع، وهو أمر بالغ الأهمية لتشخيص الأخطاء.

- audit=0: يقوم بتعطيل نظام التوقيت لتسريع عملية الإقلاع.

- GRUB_THEME="/usr/share/grub/themes/helwan/theme.txt": يطبق سمة مخصصة لمظهر جذاب واحترافي.

- GRUB_DISABLE_OS_PROBER=false: يُمكن OS-Prober، وهو مفيد لاكتشاف أنظمة التشغيل الأخرى وتسهيل إعدادات التمهيد المزدوج.

etc/mkinitcpio.conf/

يقوم هذا الملف بتهيئة القرص الأولي (initramfs)، والذي يحتوي على الوحدات الأساسية التي يتم تحميلها قبل نظام الملفات الجذر الرئيسي.

- HOOKS=(base udev modconf kms memdisk archiso archiso_loop_mnt archiso_pxe_common archiso_pxe_nbd archiso_pxe_http archiso_pxe_nfs block filesystems keyboard): تحدد هذه الخطافات ما يتم تحميله في initramfs. خطافات archiso مخصصة لتشغيل النظام من ISO أو الإقلاع عبر PXE. يضمن خطاف keyboard دعم لوحة المفاتيح في حالات الطوارئ.

- COMPRESSION="xz": يستخدم ضغط XZ، مما يؤدي إلى حجم ISO أصغر.

etc/mkinitcpio.d/linux.preset/

يحدد هذا الملف الصورة الأساسية المستخدمة للإقلاع.

- archiso_image="/boot/initramfs-linux.img": يشير إلى صورة initramfs الرئيسية للنظام.

etc/mkinitcpio.conf.d/archiso.conf/

هذه تهيئة مخصصة لـ Archiso.

- HOOKS=(base udev microcode modconf kms memdisk archiso ...): يتضمن خطاف microcode لتحديثات التعليمات البرمجية الدقيقة للمعالج، مما يعزز الاستقرار والأمان.

- COMPRESSION="xz" و (-9e COMPRESSION_OPTIONS=): يستخدم ضغطاً عالياً جداً لتقليل حجم ISO بشكل كبير.

etc/modprobe.d/broadcom-wl.conf/

يمنع ملف التهيئة هذا حدوث تعارضات مع برامج تشغيل Broadcom اللاسلكية.

- **# The broadcom-wl package requires some modules to be disabled ...**: تعليقات تشير إلى أن بعض الوحدات تم حظرها لتجنب التعارضات مع برنامج تشغيل Broadcom اللاسلكي الخاص. هذا يضمن توفر وظائف WiFi فوراً عند الإقلاع الأول.

2.6.5. مدير حزم pacman

يُعد مدير حزم pacman العمود الفقري لإدارة البرامج في Arch Linux ومشتقاته، مثل Helwan Linux. يمتد دوره إلى ما هو أبعد من مجرد تثبيت البرامج؛ فهو يشمل أيضاً تحديثات النظام، وإزالة الحزم غير المرغوب فيها، والتحقق من سلامة الحزم من خلال التوقيعات الرقمية.

— pacman.conf 4.1 مركز التحكم

يتولى الملف `etc/pacman.conf` مسؤولية تحديد كيفية عمل pacman. من خلاله، يتم تكوين المستودعات (Repositories) وسياسات الأمان وآليات تنزيل الحزم.

الأقسام الرئيسية:

• خيارات عامة [options]

- `HoldPkg = pacman glibc`: يضمن عدم إزالة أو تحديث الحزم الأساسية مثل `pacman` و `glibc` عن طريق الخطأ.
- `Architecture = auto`: يحدد البنية المستهدفة (مثل `x86_64` ، `arm`). يسمح خيار `auto` لـ `pacman` بتحديد ذلك تلقائياً.
- `ParallelDownloads = 5`: يسمح بتنزيل ما يصل إلى 5 حزم بشكل متزامن، مما يسرع العملية.
- `SigLevel = Required DatabaseOptional`: يحدد سياسة التحقق من التوقيع الرقمي: يجب توقيع الحزم، بينما يكون التحقق من توقيع قواعد بيانات المستودعات اختياريًا.
- `LocalFileSigLevel = Optional`: عند تثبيت الحزم من ملفات محلية، يكون التحقق من التوقيع اختياريًا.

• المستودعات [repositories]

- `[core]` و `[extra]`: المستودعات الرسمية والأساسية من Arch Linux.
- `[helwan]`: مستودع مخصص لتوزيع Helwan Linux ، والذي يتضمن حزمًا مخصصة تم ضبطها خصيصًا للتوزيع.
- `[helwan]`

- `SigLevel = Optional TrustedOnly`: هنا، نلاحظ أن التحقق من التوقيع اختياري، ولكن يجب أن تكون الحزم من مصدر موثوق.

Server = [https://helwan-linux.github.io/\\$repo/\\$arch](https://helwan-linux.github.io/$repo/$arch) : ▪

ملاحظة: يمكن إضافة مستودعات أخرى مثل multilib أو مخصصة، ولكنها معطلة افتراضياً.

4.2 خطافات — pacman الأتمتة

بالإضافة إلى تكوين `pacman.conf`، يمكن التحكم في سلوك `pacman` بعد العمليات باستخدام الخطافات (`hooks`) يحتوي الدليل `/etc/pacman.d/hooks/` على نصوص برمجية صغيرة يتم تنفيذها تلقائياً بعد تثبيت حزم معينة أو ترقيتها أو إزالتها.

أمثلة على الخطافات في: Helwan Linux

• `uncomment-mirrors.hook`

- المحفز (Trigger): يتم تنفيذه بعد تثبيت حزمة `pacman-mirrorlist` أو تحديثها.
- الإجراء (Action): يستخدم أمر `sed` لإلغاء التعليق على جميع الخوادم في `/etc/pacman.d/mirrorlist`، مما يؤدي فعلياً إلى تمكين جميع المرايا تلقائياً.
- الهدف: ضمان أن يكون للنظام قائمة مرايا نشطة فور الإقلاع من ISO.

• `zzzz99-remove-custom-hooks-from-airootfs.hook`

- المحفز (Trigger): يتم تنفيذه لأي عملية (تثبيت، ترقية، إزالة) على أي حزمة.
- الإجراء (Action): ينفذ نصاً برمجياً يحذف أي خطاف يحتوي على عبارة `"remove from airootfs"`.
- الهدف: هذه الخطافات ضرورية فقط أثناء عملية بناء ISO بيئة حية ولكن لا يجب أن تظل نشطة على النظام المثبت.

مع هذه الآلية، يمكننا تخصيص سلوك `pacman` في بيئة ISO دون التأثير على النظام بعد التثبيت.

4.3 خاتمة

- يحدد `pacman.conf` سياسات `pacman` الأمان، المستودعات، خيارات التنزيل.
- توفر الخطافات الأتمتة لعمليات ما بعد التثبيت أو الترقية.
- دمج مستودع Helwan Linux المخصص يجعل التوزيعة مكتفية ذاتياً، ومتميزة عن Arch، مع الاحتفاظ بمتانة النظام الأساسي.

2.6.6 إدارة المرايا (Mirrors)

عند استخدام **pacman**، تعتمد سرعة التنزيل وجودة الخدمة بشكل أساسي على اختيار الخادم (المرآة) المناسب. يوفر **Helwan Linux** آلية جاهزة لاختيار أفضل المرايا باستخدام أداة **reflector**.

2.6.6.1 ملف تكوين reflector

الموقع `/etc/xdg/reflector/reflector.conf`:

يحدد هذا الملف كيفية عمل أداة **reflector** عند تحديث قائمة المرايا. يحدد معايير مثل:

- **country**-- يحدد البلد أو مجموعة البلدان الأقرب للمستخدم.
- **protocol**-- بروتوكول التنزيل (`http/https`).
- **sort**-- كيفية ترتيب الخوادم (على سبيل المثال، حسب السرعة).
- **latest N**-- يستخدم أحدث **N** خوادم تم اختبارها.

مثال واقعي:

```
--country Egypt,France,Germany \
```

```
--protocol https \
```

```
--latest 20 \
```

```
--sort rate \
```

```
--save /etc/pacman.d/mirrorlist
```

يضمن هذا تحديث **mirrorlist** باستمرار بأفضل الخوادم، مما يسرع تنزيل الحزم بشكل كبير.

2.6.6.2 خدمة reflector في systemd

الموقع `/etc/systemd/system/reflector.service`:

تم دمج هذه الخدمة في **Helwan Linux** لتعمل عند كل إقلاع للنظام أو بشكل دوري عبر مؤقت. وظيفتها هي:

- تنفيذ أداة **reflector** باستخدام ملف التكوين المذكور أعلاه.
- تحديث `/etc/pacman.d/mirrorlist` تلقائيًا.
- ضمان أن **pacman** يستخدم دائمًا أسرع المرايا وأحدثها دون تدخل يدوي.

ميزة قوية للمستخدمين النهائيين: لا يحتاج المستخدمون إلى إدخال وتعديل المرايا يدويًا، وهو ما كان غالبًا نقطة إحباط للمبتدئين على

Arch Linux.

2.6.6.3 تكامل خدمات pacman وخطافاتها (Hooks)

لا يعتمد Helwan Linux فقط على pacman.conf والمرايا، ولكنه يستخدم أيضًا مزيجًا من خدمات systemd وخطافات pacman.

- خطافات (pacman كما رأينا سابقًا) تنظف وتعديل التكوينات أثناء تثبيت الحزم أو ترقيتها أو إزالتها.
- خدمة reflector تقوم بتحديث المرايا تلقائيًا.

يضمن هذا الإعداد أن يظل النظام سريعًا وآمنًا ومُكوّنًا من اليوم الأول.

2.6.6.4 الخلاصة

- pacman.conf: مركز التحكم لمدير الحزم.
 - خطافات: pacman: أتمتة ذكية أثناء التثبيت/الترقيات.
 - reflector: اختيار تلقائي لأفضل الخوادم.
 - خدمات: systemd: تضمن تحديث المرايا المستمر دون تدخل المستخدم.
- يجمع Helwan Linux كل هذه العناصر لتقديم نسخة جاهزة للاستخدام من Arch Linux ، مما يلغي التكوين اليدوي الشاق الذي غالبًا ما يردع معظم المستخدمين.

pacman-init.service 2.6.7 تهيئة pacman عند أول إقلاع

الموقع /etc/systemd/system/pacman-init.service :

2.6.7.1 الغرض من الخدمة

يعتمد مدير الحزم pacman على آلية مفاتيح GPG لضمان أن الحزم المضافة أو المثبتة موقعة رقميًا وتأتي من مصادر موثوقة (مستودعات Arch Linux أو Helwan). ومع ذلك، في صورة ISO أو تثبيت جديد، لا يتم تهيئة مخزن المفاتيح.

هنا يأتي دور هذه الخدمة:

- تقوم بتهيئة مخزن المفاتيح (تنشئ قاعدة بيانات المفاتيح للمرة الأولى).
- تملأ مخزن المفاتيح بمفاتيح الكيانات الموثوقة (مُعني الحزم الرسميين في Arch + مستودع Helwan).

2.6.7.2 شرح أقسام الملف

[Unit]

- Description: يوضح الغرض (تهيئة مخزن المفاتيح).
- Requires=etc-pacman.d-gnupg.mount: تعتمد الخدمة على دليل /etc/pacman.d/gnupg/، الذي يخزن المفاتيح. يجب أن يكون متاحًا قبل التنفيذ.

- `After=... time-sync.target:` تعتمد على مزامنة الوقت لأن التحقق من التوقيع الرقمي يتطلب طابعًا زمنيًا صحيحًا (لصلاحيّة الشهادة).

- `Before=archlinux-keyring-wkd-sync.service:` يجب أن تعمل قبل خدمة مزامنة المفاتيح المساعدة.

[Service]

- `Type=oneshot:` تعمل الخدمة مرة واحدة فقط أثناء الإقلاع.
- `RemainAfterExit=yes:` تعتبر نفسها نشطة حتى بعد الانتهاء، لذلك ترى الخدمات اللاحقة أن المفتاح قد تم تهيئته.
- `ExecStart=/usr/bin/pacman-key --init:` يقوم بإنشاء مخزن مفاتيح جديد.
- `ExecStart=/usr/bin/pacman-key --populate:` يضيف مفاتيح Arch و Helwan إلى المتجر.

[Install]

- `WantedBy=multi-user.target:` هذا يعني أنه سيتم تمكينه افتراضيًا في وضع التشغيل القياسي (متعدد المستخدمين).

2.6.7.3 لماذا هذا مهم؟

بدون تهيئة مخزن المفاتيح، سيرفض `pacman` تثبيت أو تحديث أي حزم موقعة. هذا يحمي النظام من أي عبث أو تنزيل لحزم غير موثوق بها. وجود هذه الخدمة يجعل صورة Helwan Linux ISO جاهزة من أول إقلاع، دون أن يضطر المستخدم إلى تنفيذ أوامر يدوية مثل:

`pacman-key --init`

`pacman-key --populate`

2.6.7.4 القيمة المضافة في Helwan Linux

في توزيع Arch القياسي، يجب على المستخدم إدارة عملية تهيئة المفاتيح بنفسه. في Helwan Linux، تم دمج هذه الخدمة "خارج الصندوق" لـ:

- توفير وقت الإعداد.
- ضمان الأمان الكامل من اليوم الأول.
- توفير تجربة سلسلة، حتى لغير الخبراء.

2.6.8 ملف reflector.conf في /etc/xdg/reflector/

هذا الملف هو جوهر خدمة reflector، الأداة المسؤولة عن تحديث ملف mirrorlist الخاص بـ pacman بشكل دوري لاختيار أسرع الخوادم المتاحة. في توزيعه مبنية باستخدام archiso مثل Helwan Linux، يضمن هذا الملف حصول المستخدمين على تجربة تنزيل سريعة ومستقرة منذ اللحظة الأولى.

محتويات الملف:

#.Reflector configuration file for the systemd service

save /etc/pacman.d/mirrorlist--

ipv4--

ipv6--

protocol https--

latest 20--

sort rate--

شرح تفصيلي:

- `save /etc/pacman.d/mirrorlist--`: أي تحديثات لخوادم Arch Linux أو Helwan Linux يتم حفظها مباشرة في هذا الملف. بعبارة أخرى، سيقراً pacman دائماً من قائمة الخوادم (mirrorlist) التي تم إنشاؤها تلقائياً. هذا يوفر للمستخدمين سرعة في تثبيت وتحديث الحزم من أقرب وأسرع الخوادم.
- `ipv4--` و `ipv6--`: يتيح هذان الخياران لـ reflector جلب الخوادم التي تدعم كلا البروتوكولين. هذا أمر بالغ الأهمية لكي تعمل التوزيع في بيئات مختلفة، سواء كانت شبكة محلية قديمة (IPv4) أو شبكات وخوادم عالمية حديثة (IPv6).
- `protocol https--`: يجبر هذا الخيار reflector على استخدام خوادم آمنة (HTTPS). الهدف هو الحماية من هجمات "الرجل في المنتصف" (MITM) وضمان تنزيل الحزم بشكل آمن من خوادم موثوقة.
- `latest 20--`: يعني هذا أن reflector سيجلب أحدث 20 خادمًا تم تحديثها من قائمة الخوادم الرسمية. الخوادم التي لم يتم تحديثها يتم إزالتها تلقائياً.
- `sort rate--`: يحدد هذا الخيار أسرع الخوادم بناءً على معدل التنزيل الخاص بها. هذا يضمن أن مستخدمي Helwan Linux يجدون تحديثاتهم بشكل أسرع بكثير من الاعتماد على قائمة خوادم عشوائية.

أهمية الملف داخل التوزيع: 7

- يخلق تجربة مستخدم احترافية: بمجرد أن يبدأ المستخدم في تشغيل البيئة الحية (Live environment) أو بعد التثبيت، فإنه يختبر سرعة عالية جداً في تنزيلات الحزم.
- يقلل المشاكل: يقلل من أخطاء "انتهاء المهلة" (time out) أو مشاكل الخوادم البطيئة التي يمكن أن تعطي انطباعاً سيئاً عن التوزيع.

- يوفر إعدادات افتراضية ذكية: يمنح Helwan Linux تكويناً ذكياً من البداية، بدلاً من إجبار المستخدم على البحث يدوياً عن الخوادم.

LightDM 2.6.9: مدير العرض الرسومي LightDM هو المكون المسؤول عن شاشة تسجيل الدخول في توزيع Helwan Linux. ملف التكوين الرئيسي الخاص به يوجد في المسار `/etc/lightdm/lightdm.conf`: هذا هو الملف الأساسي الذي يتحكم في كيفية عمل LightDM ، بما في ذلك أداة الترحيب المستخدمة (greeter) ، وبيئة سطح المكتب الافتراضية، وإعدادات تسجيل الدخول التلقائي، وكيفية التعامل مع الضيوف والجلسات. كما أنه يحتوي على خيارات متقدمة مثل دعم XDMCP و VNC. الأقسام الرئيسية في `lightdm.conf` [LightDM]

- `run-directory=/run/lightdm:` يحدد المسار الذي يستخدمه LightDM لتخزين بيانات التشغيل (ملفات PID ، مأخذ التوصيل).
 - `log-directory` (و `cache-directory` معطلة افتراضياً): تتحكم في مكان تخزين ملفات السجل والملفات المؤقتة. هذه الإعدادات مهمة لتصحيح أخطاء شاشة تسجيل الدخول "Seat:*" [Seat:*]. يشير إلى جلسة مرتبطة بشاشة عرض. الإعدادات الأكثر أهمية هي:
 - `greeter-session=lightdm-slick-greeter:` يحدد برنامج "الترحيب" الذي سيتم استخدامه (في هذه الحالة، -slick-greeter).
 - `user-session=cinnamon:` يحدد بيئة سطح المكتب الافتراضية بعد أن يقوم المستخدم بتسجيل الدخول. (Cinnamon).
 - `session-wrapper=/etc/lightdm/Xsession:` وسيط يستخدمه LightDM لإطلاق جلسة المستخدم. تشمل الإعدادات الأخرى:
 - `autologin-user:` يُمكن تسجيل الدخول التلقائي لمستخدم محدد.
 - `greeter-hide-users:` يخفي قائمة المستخدمين.
 - `allow-guest:` يسمح بوجود حساب ضيف. هذه المرونة تجعل الملف قابلاً للتخصيص لمحطة عمل واحدة أو لنظام متعدد المقاعد [XDMCP Server]. هذا القسم يسمح بالاتصالات عن بعد باستخدام بروتوكول XDMCP وهو معطل افتراضياً (enabled=false) لأسباب أمنية [VNC Server]. يُمكن هذا القسم تسجيل الدخول عن بعد عبر VNC وهو معطل أيضاً افتراضياً. يمكن أن يكون مفيداً في بيئات الدعم الفني أو الفصول الدراسية.
- 2.6.9.1 slick-greeter:** تخصيص شاشة تسجيل الدخول ملف التكوين الخاص بـ slick-greeter يوجد في المسار `/etc/lightdm/slick-greeter.conf` هذا الملف مسؤول عن المظهر والتجربة الرسومية لشاشة تسجيل الدخول. إعدادات مهمة في [Greeter]
- `background=/usr/share/backgrounds/login.png:` الخلفية الأساسية لشاشة تسجيل الدخول.
 - `theme-name=Arc-Dark:` النسق المستخدم على شاشة تسجيل الدخول.
 - `icon-theme-name=Qogir:` نسق الأيقونات الافتراضي.
 - `cursor-theme-name=Qogir` و `cursor-theme-size=16:` مظهر وحجم مؤشر الماوس.

- draw-user-backgrounds=false: يمنع استخدام خلفية شخصية لكل مستخدم.
- show-power=false: و show-a11y=false يخفي خيارات الوصول والتحكم بالطاقة من شاشة تسجيل الدخول.
- background-color=#000000: لون خلفية بديل (في حالة فشل تحميل الصورة). (ملخص
- lightdm.conf: هذا هو الملف المركزي الذي يحدد كيفية عمل LightDM الجلسات، تسجيل الدخول التلقائي، البروتوكولات عن بعد).
- slick-greeter.conf: هذا الملف مخصص لمظهر شاشة تسجيل الدخول (الخلفية، النسق، الأيقونات). (يوضح هذا الفصل بوضوح الفصل بين الوظائف والمظهر:
- الوظائف موجودة في lightdm.conf.
- المظهر موجود في slick-greeter.conf.
- 2.6.10 ملفات تكوين الشبكة etc/resolv.conf/
- الغرض: يحدد خوادم DNS التي يستخدمها النظام لحل الأسماء.
- المحتوى النموذجي:
- nameserver 1.1.1.1
- nameserver 8.8.8.8
- ملاحظات Helwan/Archiso: في البيئة الحية، تتم إدارة هذا الملف عادةً بواسطة systemd-resolved أو يتم إنشاؤه بواسطة dhcpcd/NetworkManager. تجنب كتابة إدخال DNS ثابتة إذا كان النظام يستخدم systemd-resolved. دع resolv.conf يتم إنشاؤه تلقائيًا أو أنشئ رابطًا رمزيًا إلى /run/systemd/resolve/stub-resolv.conf.
- etc/systemd/network/20-ethernet.network/
- الغرض: يحدد سياسات الشبكة لواجهات Ethernet (ثابتة أو DHCP).
- المحتوى النموذجي:

[Match]

*en=Name

[Network]

yes=DHCP

- ملاحظات: يستخدم ISO DHCP لتبسيط الوصول إلى الشبكة عبر أجهزة مختلفة. للحصول على عنوان IP ثابت أثناء التثبيت، يشرح الدليل كيفية تغيير DHCP=no وإضافة إعدادات Address و Gateway و DNS.=

etc/systemd/network/20-wlan.network/

- الغرض: قواعد لواجهات Wi-Fi (*wlan).

- المحتوى النموذجي:

[Match]

*wlan=Name

[Network]

yes=DHCP

- ملاحظات: عادةً ما يتم إبقاؤه بسيطاً في البيئات الحية، حيث تتعامل أدوات مثل NetworkManager أو iwd مع اتصالات Wi-Fi الفعلية. تتضمن ملفات network هذه حصول واجهات wlan على عناوين DHCP بمجرد توصيلها. اشرح العلاقة بين iwd أو wpa_supplicant وهذا الملف، إن أمكن.

etc/systemd/network/20-wwan.network/

- الغرض: إعدادات لواجهات الاتصال الخلوي (WWAN).
- المحتوى النموذجي: مشابه لـ wlan ولكنه قد يتضمن IPv6AcceptRA=no أو تكوينات خاصة بالموجهات.
- ملاحظات: مهم للأجهزة التي تستخدم بيانات خلوية. في البيئات الحية، تُترك عادةً لمديري الشبكات مثل ModemManager.

etc/systemd/network.conf.d/ipv6-privacy-extensions.conf/

- الغرض: يقوم بتكوين سياسة خصوصية (RFC 4941 IPv6) عن طريق إنشاء عناوين مؤقتة لتقليل تتبع الجهاز.
- المحتوى النموذجي:

[Network]

1=PrivacyExtensions

- ملاحظات: مفيد لخصوصية المستخدم. في بعض الشبكات، قد يلزم تعطيله (0) لأسباب التوافق. اذكر تأثيره: خصوصية محسنة مقابل صعوبة تتبع واجهة ثابتة.

etc/systemd/resolved.conf.d/archiso.conf/

- الغرض: إعدادات إضافية لـ systemd-resolved داخل بيئة ISO.
- محتوى نموذجي سياقي:

[Resolve]

1.1.1.1=DNS

8.8.8.8

9.9.9.9=FallbackDNS

yes=Cache

no=DNSOverTLS

- ملاحظات Helwan: في البيئة الحية، من المرغوب توفير خوادم DNS معروفة ومستقرة (مثل Cloudflare/Google) لتجنب الاستثناءات أثناء الإعداد الأولي للمستخدم. ومع ذلك، بعد التثبيت، يُوصى بترك المستخدم يختار أو استخدام NetworkManager/reflector لتحديث الإعدادات.

etc/ssh/sshd_config.d/10-archiso.conf/

- الغرض: إعدادات SSH مخصصة لبيئة ISO/الحية.
- أمثلة على الخيارات المتوقعة:

• Allow SSH for remote troubleshooting in live environment #

• PermitRootLogin=yes

• PasswordAuthentication=yes

• or maybe a restricted setting: PermitRootLogin=prohibit-password #

- ملاحظات أمنية: في البيئات الحية، يمكن تمكين SSH لتسهيل الدعم عن بعد. ومع ذلك، من الضروري دائمًا ذكر أن هذه الإعدادات غير مناسبة لنظام مثبت في التوثيق. بعد التثبيت، يُفضل تعطيل PermitRootLogin أو تقييد الوصول إلى المصادقة المستندة إلى المفاتيح فقط.

ملخص سريع (عملي، بلا حشو)

resolv.conf يتحكم في نظام أسماء النطاقات (DNS) في بيئة النظام الحي (Live Environment)، من الأفضل ترك إدارته لـ systemd-resolved أو NetworkManager.

systemd/network/20-*.network: ملفات تضمن تفعيل DHCP التلقائي للواجهات الشبكية في بيئة النظام الحي.

ip6-privacy-extensions.conf: يدير خصوصية IPv6. يشرح الخيار وتأثيراته.

resolved.conf.d/archiso.conf: يوفر نظام DNS ثابتًا وآمنًا في بيئة النظام الحي لتجربة موثوقة.

sshd_config.d/10-archiso.conf: يمكن SSH خصيصًا لبيئة النظام الحي؛ ويجب تحذير المستخدمين من تركه مفعلاً على نظام مثبت.

livecd-alsa-unmuter.service

الهدف: خدمة مصممة لإزالة كتم صوت أجهزة ALSA افتراضياً في بيئة النظام الحي. هذا ضروري لأن بعض بطاقات الصوت في لينكس تبدأ في حالة "صامتة (muted)"، مما قد يربك المستخدمين ويجعلهم يظنون أن الصوت لا يعمل.

المحتوى النموذجي:

[Unit]

Description=Unmute ALSA sound devices in Live CD

After=sound.target

[Service]

Type=oneshot

ExecStart=/usr/bin/alsactl init

ExecStart=/usr/bin/amixer -c 0 set Master unmute

ExecStart=/usr/bin/amixer -c 0 set PCM unmute

[Install]

WantedBy=multi-user.target

شرح:

- ExecStart=/usr/bin/alsactl init: يُهيئ إعدادات صوت ALSA.
 - amixer set Master/PCM unmute: يزيل كتم الصوت عن القنوات الصوتية الرئيسية (Master/PCM).
 - WantedBy=multi-user.target: الخدمة تبدأ تلقائيًا مع النظام في بيئة النظام الحي.
- ملاحظات Helwan: هذا الملف مفيد لأنه يضمن تجربة سلسلة: يبدأ النظام وصوت يعمل دون الحاجة إلى إعدادات يدوية. بعد التثبيت على القرص الصلب، يُفضل ترك إعدادات ALSA أو PulseAudio/PipeWire لتفضيلات المستخدم.

getty@tty1.service.d/autologin.conf

الهدف: يمكن تسجيل الدخول التلقائي للمستخدم liveuser على طرفية TTY1 عند بدء تشغيل النظام الحي.

المحتوى النموذجي:

[Service]

ExecStart=

ExecStart=-/sbin/agetty --autologin liveuser --noclear %I \$TERM

شرح:

- ExecStart= (empty line): سطر فارغ يمسح القيمة الأصلية.
 - --autologin liveuser: getsyجل دخول liveuser مباشرة دون الحاجة لكلمة مرور.
 - --noclear: يمنع مسح شاشة الطرفية بعد تسجيل الدخول.
- ملاحظات Helwan: هذه ميزة ضرورية لسهولة الاستخدام، خاصة للمبتدئين. بعد التثبيت، يتم تعطيل هذا الإعداد، ويعود السلوك الطبيعي (إدخال المستخدم كلمة المرور لتسجيل الدخول).

display-manager.service

الهدف: خدمة عامة تحدد أي مدير عرض (Display Manager) سيتم تشغيله على النظام. في Helwan Linux ، تم ربطها بـ LightDM.

المحتوى النموذجي:

[Unit]

Description=Display Manager

Conflicts=getty@tty1.service

After=systemd-user-sessions.service getty@tty1.service

[Service]

ExecStart=/usr/bin/lightdm

Restart=always

[Install]

Alias=display-manager.service

WantedBy=graphical.target

شرح:

- Conflicts=getty@tty1.service: يمنع getsyمن العمل بالتزامن مع LightDM.
- ExecStart=/usr/bin/lightdm: هنا، كان الاختيار هو (LightDM مع slick-greeter).
- WantedBy=graphical.target: يضمن بدء LightDM عند الدخول إلى الوضع الرسومي.

ملاحظات Helwan: ربط الخدمة بـ LightDM يضيف تجربة مستخدم رسومية مباشرة. بعد التثبيت، يمكن للمستخدمين التغيير إلى أي مدير عرض آخر (GDM ، SDDM...).

choose-mirror.service

الهدف: يختار تلقائيًا أسرع مستودعات (mirrors) عند بدء تشغيل بيئة النظام الحي. يهدف هذا إلى تحسين سرعة تنزيل الحزم وكفاءة تثبيت النظام.

المحتوى النموذجي:

[Unit]

Description=Choose the fastest pacman mirror

After=network-online.target

Wants=network-online.target

[Service]

Type=oneshot

ExecStart=/usr/bin/reflectord --protocol https --latest 20 --sort rate --save /etc/pacman.d/mirrorlist

[Install]

WantedBy=multi-user.target

شرح:

- After=network-online.target: الخدمة تنتظر حتى يصبح الاتصال بالشبكة جاهزًا بالكامل قبل أن تبدأ.
- ExecStart=reflectord: يستخدم أداة Reflectord لتحديد أسرع 20 خادمًا HTTPS ، ويرتبها حسب سرعة الاتصال.
- WantedBy=multi-user.target: بضمن بدء هذه الخدمة تلقائيًا عند الدخول إلى وضع المستخدمين المتعددين (multi-user mode).

ملاحظات Helwan: هذه الخطوة عيقرية لأنها تقلل بشكل كبير من وقت التثبيت وتوفر تجربة سلسلة، حتى للمستخدمين البعيدين جغرافيًا عن الخوادم الأساسية لـ Arch Linux.

reflector.service

الهدف: يحدّث المستودعات بشكل دوري بناءً على الإعدادات المحددة في `/etc/xdg/reflector/reflector.conf`.

المحتوى النموذجي:

[Unit]

Description=Pacman mirrorlist update

[Service]

Type=oneshot

ExecStart=/usr/bin/reflector --config /etc/xdg/reflector/reflector.conf

شرح:

- **reflector --config:** يقرأ الإعدادات من الملف المحدد ويحدّث قائمة المستودعات. يمكن تشغيل هذه الخدمة يدوياً أو ربطها بخدمة مؤقتة (timer) للتحديثات المجدولة.

ملاحظات Helwan: هذا يضمن أن النظام لا يظل مرتبطاً بخادم بطيء بشكل دائم، مما يضيف مرونة وسرعة على المدى الطويل.

reflector.timer

الهدف: يحدد جدول تشغيل خدمة `reflector.service`.

المحتوى النموذجي:

[Unit]

Description=Run reflector weekly

[Timer]

OnBootSec=10min

OnUnitActiveSec=1w

[Install]

WantedBy=timers.target

شرح:

- `OnBootSec=10min`: التشغيل الأولي يحدث بعد 10 دقائق من بدء التشغيل.
 - `OnUnitActiveSec=1w`: التحديثات اللاحقة ستُجرى تلقائياً مرة واحدة كل أسبوع.
 - `WantedBy=timers.target`: يسجل هذا الموقت ضمن نظام إدارة الموقتات في `systemd`.
- ملاحظات Helwan: هذا حل ذكي يضمن تحديث المستودعات دون تدخل يدوي. يمكن للمستخدم نسيان الأمر تماماً، وسيقوم النظام بإدارة نفسه.

systemd-networkd.service

الهدف: يدير الاتصال بالشبكة باستخدام `systemd-networkd` بدلاً من `NetworkManager`. هذا الخيار مناسب للبيئات الخفيفة أو أنظمة التشغيل الحية. (Live systems)
المحتوى النموذجي:

[Unit]

Description=Network Service

Documentation=man:systemd-networkd.service(8)

ConditionCapability=CAP_NET_ADMIN

After=network-pre.target

Before=network.target

Wants=network.target

[Service]

ExecStart=/usr/lib/systemd/systemd-networkd

Restart=always

[Install]

WantedBy=multi-user.target

شرح:

- `ExecStart`: يشغل الـ `daemon` المسؤول عن إدارة الشبكة (الواجهات، DHCP، الإعدادات الثابتة...).

- `After=network-pre.target` و `Before=network.target` يضمن الترتيب الصحيح لتجهيز الشبكة قبل استخدامها.
- `Restart=always:` الخدمة تُعاد تشغيلها تلقائيًا إذا فشلت.

ملاحظات `Helwan: systemd-networkd` خيار ممتاز لـ `ISO` النظام الحي لأنه خفيف الوزن وسريع ولا يتطلب واجهة رسومية. بعد التثبيت، يمكن للمستخدم استبداله بـ `NetworkManager` إذا كان يفضل واجهة رسومية.