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 دائمًا يكمن في التكنولوجيا. هذا الشغف دفعني باستمرار لاستكشاف عالم الحوسبة والبرمجيات الحرة، ودائمًا ما كنت أسأل نفسي: **"**كيف يعمل هذا؟**"** و**"**هل يمكنني بناء شيء أفضل؟**."**

هذا الفضول المستمر قادني عميقًا إلى عالم أنظمة التشغيل، حيث اكتشفت فلسفة **آرش لينكس** —فلسفة تتماشى تمامًا 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 توزيعة تعكس هوية المجتمع العربي، وتقدم تجربة مستخدم سلسة وشخصية، وتقف كمثال ملهم لما يمكن تحقيقه بالشغف والمعرفة.

على مر السنين، قمت بتوثيق هذه الرحلة على قناتيSMA Coding ، ساعيًا لمشاركة كل ما تعلمته مع المبرمجين وعشاق لينكس في كل مكان. هذا الكتاب هو تتويج لتلك الرحلة: ليس مجرد دليل تقني، بل هو خلاصة سنوات من التجربة والخطأ، والتعلم، والمشاركة. آمل أن تجدوا في هذه الصفحات كل ما تحتاجونه لتحويل أفكاركم إلى واقع — وأن تلهمكم لتلعبوا دورًا 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 يُورّث .

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 ونظامUnix ، اللذان وضعا الأساس لكل ما نستخدمه اليوم.
- **ريتشارد ستالمان**، فيلسوف البرمجيات الحرة، الذي زرع بذور الحركة.
- **لينوس تورفالدس**، الذي جعل لينكس حقيقة ووحدنا 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.
- **الهاكرز :(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 توزيعة 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 بالجذور 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 لآرش لينكس :(Arch Linux)** لن تتعلم فقط كيفية استخدام آرش، بل ستفهم الفلسفة التي بني عليها. ستكتشف مبدأ "التحكم الكامل" وكيف يمكن لهذا الأساس المتين أن يكون لوحة قماشية فارغة لمشروعك، مما يحررك 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 بيئات سطح المكتب، وكتابة نصوص برمجية مخصصة لأتمتة مهامك.
- **تحكم كامل في :Calamares** ستفهم كيفية دمج واجهة تثبيت رسومية سلسة واحترافية في توزيعتك. ستتعمق في تخصيص كل تفصيل فيCalamares ، 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 توزيعة لينكس عملية، مستقرة، وجاهزة للاستخدام. ستتجاوز 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 آفاقًا جديدة للتحكم والتخصيص. ستنتقل 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 بسيطًا، أيها الغبي :Keep It Simple, Stupid -** يتبنى آرتش مبدأ "البساطة". هذا يعني أنه يتجنب التعقيد غير الضروري، مما يجعله شفافًا 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 مسبقًا، 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 دائمًا 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 دائمًا 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 سلسة ومخصصة للمستخدمين في المنطقة العربية. من خلال تحليل هذا المشروع، ستتعلم كيفية تطبيق المفاهيم التي سنتعلمها، وكيف يمكن للشغف الجماعي أن يخلق شيئًا ذا تأثير حقيقي. حلوان لينكس ليست مجرد توزيعة؛ إنها شهادة حية على أن الإبداع لا يعرف حدودًا 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

1.1.1نشأة نظام لينكس (1991) في أوائل التسعينيات، كان عالم أنظمة التشغيل تهيمن عليه أنظمة يونكس التجارية ونظام-MS DOSمن 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) " .(Kernelعندما تم دمجها مع أدوات مشروع جنو (الذي أسسه ريتشارد ستولمان)، حصلنا على ما نعرفه اليوم بنظام التشغيل جنو/لينكس.(GNU/Linux)

1.1.2بداية 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 توزيعة فلسفتها الخاصة: البعض ركز على سهولة الاستخدام (مثل أوبونتو لاحقًا)، والبعض على الاستقرار (مثل دبيان)، والبعض الآخر على التخصيص (مثل جنتو)

1.1.3ظهور 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 جنتو، 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.

1.1.4المجتمع 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 و.Manjaro**

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 جنتو، وليس مقيدًا مثل أوبونتو.**
- **اليوم، آرتش ليس مجرد توزيعة، 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 بتثبيت 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.

1.2.3المرونة(Flexibility)

آرتش لا يفرض عليك أي قرارات:

- **هل تريد استخدامKDE Plasma ؟ يمكنك تثبيته.**
- **هل تفضل GNOME أوXFCE ؟ كلاهما ممكن.**
- **لا تريد واجهة رسومية على الإطلاق وتفضل مدير نوافذ بسيط مثل i3wm أوbspwm ؟ كل ذلك متاح بالكامل.**

هذه المرونة تجعل آرتش مناسبًا لمختلف المستخدمين، من عشاق السرعة والبصمة الصغيرة إلى المتحمسين للأشكال والواجهات المتقدمة.

1.2.4الإصدار 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.**

1.2.5مدير الحزمPacman

أحد ركائز آرتش لينكس هو مدير الحزم .pacman

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

أمثلة شائعة:

Bash

#تحديث النظام

sudo pacman -Syu

#تثبيت برنامج

sudo pacman -S firefox

#إزالة برنامج مع ملفاته الإضافية

sudo pacman -Rns firefox

#البحث عن برنامج

pacman -Ss vlc

مقارنة:

- **aptفي دبيان/أوبونتو قد يتطلب أوامر أكثر تعقيدًا.**
- **dnfفي فيدورا أبطأ أحيانًا في الأداء.**

1.2.6 Arch Wiki: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 رسوميات 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 PKGBUILDالتي 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 مثل :yay

Bash

yay -S google-chrome

ولكن يجب أن تكون حذرًا:

- ليست كل الحزم في AUR مضمونة الجودة العالية أو الأمان.
- يوصى دائمًا بقراءة ملف PKGBUILDقبل التثبيت.

1.2.8آرتش 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 الرسوميات 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 توزيعة 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 KISS (مبدأ) ابقها بسيطة، أيها الغبي

كلمة "بساطة" هنا لا تعني سهولة الاستخدام. آرتش لا يحاول أن يكون سهلًا 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 بتثبيت تعريف لبطاقة الرسوميات، توجد أدوات رسومية مخصصة تقوم بالمهمة نيابة عنك.
  - في آرتش، يتم ذلك عن طريق تثبيت الحزم المناسبة يدويًا من pacmanأو 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)

يُبنى 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 أردتGNOME ، 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 ومفهومًا.

- 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.

1.3.4التعلم بالممارسة

يختلف آرتش لينكس عن معظم التوزيعات لأنه يعلّمك 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 يعلّمك كيف تصطاد".

1.3.5نموذج 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 ئة 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.

1.3.6التطور المجتمعي

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

- **المجتمع هو القلب النابض للتوزيعة.**
- **معظم الحلول تجدها فيArch Wiki ، الذي كتبه المستخدمون.**
- **مستودع AUR مبني بالكامل على مساهمات المجتمع.**

يعكس هذا مبدأً فلسفيًا هامًا: المعرفة جماعية وليست حكرًا على مؤسسة أو شركة.

1.3.7آرتش 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 توزيعة: إنه طريقة تفكير وتعامل مع أنظمة التشغيل.

1.4نموذج التحديث المستمر(Rolling Release)

إحدى أبرز ميزات آرتش لينكس، وما يجعله مختلفًا 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.

1.4.1ما 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 دبيان 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 فورًا في المستودعات.
  - مثال: آرتش لينكس يمتلك دائمًا أحدث إصدار من نواة لينكس، دون انتظار "إصدار جديد من آرتش."

1.4.2مميزات التحديث المستمر

- دائمًا أحدث البرامج :تحصل على أحدث إصدار من محرر النصوص المفضل لديك أو بيئة البرمجة بمجرد صدورها. هذا مناسب جدًا 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 أبدًا. طالما أنك تقوم بتحديثه، فإنه سيظل دائمًا في أحدث صورة له.

1.4.3عيوب 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 الرسوميات أحيانًا في مشاكل في الإقلاع.
- مسؤولية أكبر على المستخدم :يجب عليك متابعة أخبار آرتش باستمرار للتأكد من وجود أي تنبيهات مهمة قبل التحديث، وأن تكون مستعدًا لإصلاح المشكلات بنفسك.

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

1.4.4كيفية 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**

1.4.5استراتيجيات 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 -Syuبسيطًا، لكن يجب أن تكون على دراية بما قد يتغير بعده.**

1.4.6مقارنة مع توزيعات أخرى

فيدورا (شبه مستمر)	أوبونتو/دبيان (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	6أشهر أو أكثر	لا يوجد (مستمر)	الفاصل بين الإصدارات
متوسط	أسهل للمبتدئين	يحتاج إلى متابعة مستمرة	سهولة الإدارة

1.4.7أمثلة عملية

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

خلاصة القسم 1.4 :نموذج التحديث المستمر في آرتش لينكس هو ميزة قوية تجعله دائمًا عصريًا ومواكبًا. لكنه سلاح ذو حدين: يمنحك الحرية والحداثة، ولكنه يتطلب مسؤولية ويقظة.

1.5مدير الحزمPacman

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 التوزيعة. لكن Pacman يبرز بفضل فلسفته البسيطة وأدائه القوي.

1.5.1ما هوPacman ؟

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

1.5.2مميزاتPacman

- 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.(

1.5.3أوامر Pacman الأساسية

الأمر	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 محليًا	pacman -Qs python
pacman -Qi package	عرض معلومات عن برنامج مثبت	pacman -Qi nano
pacman -Syu	تحديث النظام بالكامل	pacman -Syu

1.5.4ملفات إعدادPacman

- **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
```

1.5.5التوقيعات 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 بالتثبيت إلا إذا أجبرت النظام (وهو أمر غير مستحسن.(**

1.5.6مقارنة 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 منPacman

1.5.7استخدام 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 بتشغيلSyu- sudo pacman :

1.5.8مشاكل شائعة مع Pacman وحلولها

- قاعدة بيانات تالفة:
  - استخدم) sudo pacman -Syyلإجبار 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.

1.5.9العلاقة بين Pacman وAUR

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

- **بالنسبة لحزمAUR ، تستخدم أدوات مساعدة مثل yayأو 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.

1.6مستودع مستخدمي آرتش(AUR)

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

1.6.1ما هوAUR ؟

- AURهو اختصار لـ (Arch User Repositoryمستودع 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. أماAUR ، فهو مساحة مفتوحة حيث يمكن للمستخدمين مشاركة أي برنامج أو أداة أو حتى سمة.

1.6.2ما هو ملفPKGBUILD ؟

- ملف PKGBUILD هو برنامج نصي نصي مكتوب بلغة.Bash
- يحتوي على تعليمات لبناء حزمة من المصدر أو من ملفات مجمعة مسبقًا.
- Pacmanلا 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 مهمًا؟

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

1.6.4 كيف يعمل AUR عمليًا؟

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

1.6.5أدوات مساعدة لـAUR

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 سيبحث فيAUR ، ويبني الحزمة، ويقوم بتثبيتها تلقائيًا.

1.6.6تحديات ومخاطرAUR

- 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.

1.6.7العلاقة 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.

1.6.8مقارنة AUR مع مستودعات مشابهة في توزيعات أخرى

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

1.6.9أمثلة 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 آرتش 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.

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

أحد أهم ركائز آرتش لينكس هو فلسفةKISS ، التي تقف اختصارًا 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.

1.7.1ما 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.

1.7.2تجليات مبدأ KISS في آرتش لينكس

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

1.7.3لماذا يعتبر KISS مهمًا 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.**

1.7.4الفرق 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 بسيطًا ولكنه شفاف.

1.7.5أمثلة عملية لمبدأ KISS في آرتش

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

1.7.6انتقادات لمفهوم KISS في آرتش

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

- منحنى تعليمي حاد :قد يجد المبتدئ الأمر صعبًا جدًا في البداية.
- عمل متكرر :أحيانًا تحتاج إلى إعداد أشياء يدوية كان من الممكن أن تكون مؤتمتة.
- وقت مستهلك :تخصيص آرتش يستغرق وقتًا 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.

1.7.7 KISSعلى 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 جاهزًا".

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

خلاصة القسم 1.7 :فلسفة KISS هي ما يجعل آرتش لينكس فريدًا بين التوزيعات. إنها فلسفة بساطة واضحة تمنح المستخدم المرونة والتحكم الكامل على حساب منحنى 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 Wikiهو 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.**

1.8.3المجتمع كمصدر للدعم

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

1.8.4روح المشاركة والتعلم

- **في آرتش، الفلسفة ليست فقطKISS ، بل أيضًا "علّم غيرك."**
- **المستخدم الجديد الذي يستفيد من المنتدى أو الويكي غالبًا ما يعود لاحقًا ليساعد الآخرين.**
- **هذه الدورة المستمرة من التعلم والمشاركة جعلت آرتش أكثر من مجرد توزيعة: إنها مدرسة في لينكس.**

1.8.5انتقادات للمجتمع

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

1.8.6دروس 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.

1.9آرتش لينكس كأساس لتوزيعات أخرى

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

1.9.1مانجارو 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.

1.9.2إنديفور 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 توقفAntergos ، وُلد مشروع EndeavourOSعام 2019 كمبادرة مجتمعية.

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

- مثبتًا رسوميًا.
- خيارات متعددة لبيئات سطح المكتبKDE) ، XFCE، GNOME، إلخ.(
- دعمًا مجتمعيًا قويًا يشبه روح آرتش نفسها.

تُعتبر خيارًا مثاليًا للمستخدم الذي يريد تجربة مطابقة تقريبًا لآرتش، ولكن مع بداية أسهل.

1.9.3حلوان لينكس: الهوية المصرية في عالم آرتش

حلوان لينكس هي توزيعة مبنية على آرتش، أُنشئت 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 توزيعة، بل هو منصة مفتوحة تسمح لأي مطور أو فريق بإنشاء مشروع جديد فوقها.

1.9.4دروس 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 توزيعة 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.

1.10حالات الاستخدام

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

1.10.1للمطورين: 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 البرمجةPython) ، Go، Rust، إلخ.(
  - المكتبات وأدوات التطوير الحديثة دون الحاجة إلى انتظار "إصدار جديد" للتوزيعة.
  - القدرة على بناء بيئة تطوير مخصصة بالكامل بناءً على احتياجات مشروعهم.

مثال :مطور يعمل على مشروع ذكاء اصطناعي سيستفيد من توفر أحدث إصدارات TensorFlowأو PyTorchفي مستودعات آرتش أو مستودع .AUR

1.10.2للباحثين 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 AURأو حتى بناء الحزم يدويًا.

هذا الأمر مهم بشكل خاص في المجالات العلمية مثل الحوسبة عالية الأداء(HPC) ، والفيزياء، والرياضيات التطبيقية.

1.10.3للمستخدم العادي: بين الفضول والتحدي

- ليس آرتش التوزيعة 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 الحداثة 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.

1.11خاتمة 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، بدلًا 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 مطورًا، ستجد أحدث الحزم والأدوات في متناول يدك.**
- **إذا كنت باحثًا، ستكون لديك القدرة على التحكم في كل تفصيلة صغيرة وكبيرة داخل بيئتك.**
- **وإذا كنت مستخدمًا عاديًا شغوفًا 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.

بانتهاء هذا الفصل التمهيدي، نكون قد وضعنا الأساس النظري لفهم آرتش لينكس. الآن، حان الوقت للانتقال إلى الفصل الثاني: إعداد بيئة بناءArchiso ، حيث سنبدأ الجانب العملي ونكتشف كيف يمكننا إعداد بيئة العمل لبناء توزيعة مخصصة بدءًا من آرتش.

الفصل الثاني: إعداد بيئة البناء باستخدامArchiso

2.1مقدمة إلىArchiso

ما هوArchiso ؟ Archisoهو الإطار الرسمي الذي طوّره مجتمع Arch Linux وصانوه لبناء صور ISO قابلة للإقلاع (Live ISO images).تخدم هذه الصور أغراضًا متنوعة، بما في ذلك توزيعة 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 توزيعة 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 السكربتات وملفات الإعدادات التي تمكنك من توليد نظام تشغيل قابل للإقلاع بصيغة.ISO

لماذا نستخدمArchiso ؟ لفهم أهميته، دعنا نسأل: "ما الذي يحدد وجود توزيعة لينكس؟" الإجابة تكمن في ملف ISO متاح بسهولة، يمكن للمستخدمين تنزيله وحرقه على محرك أقراص USB والقيام بالإقلاع منه Archiso .هي الأداة التي تمكنك من إنشاء هذا الملف بنفسك.

تتضمن الأسباب الرئيسية لاستخدامه:

- **التحكم الكامل :أنت من يحدد الحزم، والإعدادات، وبيئة سطح المكتب.**
- **توزيعة شخصية :قم ببناء نظامك الشخصي وشاركه مع الآخرين.**
- **أغراض الصيانة :أنشئ ملف ISO مخصصًا لفريقك، مزودًا 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 توزيعة مثل حلوان لينكس

لتوضيح الأمر أكثر، دعنا نقارن بين سيناريوهين:

العنصر	بناء تثبيت آرتش بسيط	تخصيص توزيعة 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 ملفISO	معتدل (700 – 900 ميجابايت(	متغير: خفيف جدًا (Fluxbox) إلى ثقيل نسبيًا(Cinnamon)

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

المتطلبات الأساسية للبناء باستخدامArchiso

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

- **أجهزة مناسبة:**
  - **معالج حديث (يفضل x86_64، مع دعم virtualizationإذا كنت ستختبر على .(QEMU/VirtualBox**
  - **ذاكرة وصول عشوائي (RAM) لا تقل عن 4جيجابايت)ويوصى بـ 8جيجابايت.(**
  - **مساحة تخزين 20 :جيجابايت أو أكثر (تتطلب عملية البناء ذاكرة مؤقتة ودلائل مؤقتة.(**
  - **يوصى بشدة بقرص SSDلتسريع عمليات الضغط وفك الضغط.**
- **نظام آرتش لينكس أو نظام مشتق منه:**
  - **يجب أن يكون لديك نظام آرتش لينكس مثبت، أو توزيعة 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 sudoبدلًا من تسجيل الدخول كـ rootمباشرةً.**

ملاحظة تاريخية

Archisoليس تطورًا حديثًا.

- **كانت أصوله بسيطة جدًا: 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 تُسمى .Helwan Light**
- **ببساطة، تقوم بنسخ ملفات الإعدادات منArchiso ، وتعدّل ملف packages.x86_64، وتزيل بيئات سطح المكتب الثقيلة مثل GNOMEأو .KDE**
- **في غضون ساعة أو ساعتين، سيكون لديك ملف ISO بحجم 600 ميجابايت يُقلع إلى بيئة Fluxboxخفيفة جدًا.**

هذا يوضح كيف يبسّط Archiso العملية بشكل كبير ويوفر بيئة ملموسة للتجريب.

خاتمة (مقدمة الفصل(

- **Archisoهو المحرك الأساسي لبناء أي توزيعة مبنية على آرتش.**
- **إنه يمكنك من إعادة بناء نسخة بسيطة أو الابتكار بتوزيعة 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 توزيعتك في VirtualBox.**

2.2مقدمة إلىArchiso

تتمحور عملية بناء توزيعة مبنية على آرتش لينكس باستخدام أداة archisoحول 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 BIOSو .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 التوزيعة القدرة على تخصيص كل جزء بدقة، من الحزم المضمنة إلى واجهة المستخدم عند الإقلاع الأول.

تثبيت أداة archisoأداة archisoهي المسؤولة عن بناء صورة .ISO إنها متاحة في مستودع extraالرسمي لآرتش لينكس ويمكن تثبيتها بالأمر التالي:

Bash

```
sudo pacman -S archiso
```

ملاحظة :يُنصح بالقيام بعملية البناء داخل بيئة آرتش نظيفة (أو على الأقل داخل chrootأو (VMلتجنب التعارضات مع النظام المضيف.

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

Bash

```
cp -r /usr/share/archiso/configs/releng/ ~/archlive/
```

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

```
archlive /
├── 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 .archisoفي الأقسام التالية، سيتم شرح كل دليل وملف بالتفصيل.

releng/

هو الدليل الأساسي لملف إعدادات .archisoكل 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 سكربتات التخصيص مثل customize_airootfs.shأو ملفات الإعداد الشخصية (dotfiles) مثل bashrc.، .zshrc.**
- **usr/ ← ملفات ومكتبات إضافية للمستخدم. مثل إضافات لـ /binو /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 archisoبالدخول إلى بيئة معزولة (chroot) ومتابعة البناء.

efiboot/

يحتوي على ملفات محمل الإقلاع) EFI إما systemd-bootأو GRUB لـ .(UEFI هذا المجلد يحمل ملفات efi.والإعدادات الخاصة بمدخل الإقلاع.UEFI

grub/

إعدادات محمل الإقلاع) GRUB إذا اخترت استخدامه). يحتوي على ملفات الإقلاع لقائمةGRUB ، 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، ستُدرج:

- **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 archisoهذا الملف خطوة بخطوة لبناء ملف الـISO.**

syslinux/

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

- **← archiso_head.cfg بداية إعداد قائمة الإقلاع (العنوان، المهلة الزمنية.(**
- **← archiso_pxe-linux.cfg & archiso_pxe.cfg إعدادات الإقلاع عبر) PXEالإقلاع عبر الشبكة.(**
- **← archiso_sys-linux.cfg & archiso_sys.cfg إعدادات الإقلاع القياسية من ملف الـISO.**
- **← archiso_tail.cfg نهاية إعداد قائمة الإقلاع.**
- **← splash.png خلفية رسومية لشاشة.Syslinux**
- **← syslinux.cfg الملف الرئيسي الذي يضم ملفات الإعدادات الأخرىhead) ، tail، sys، (pxeويُعرّف قائمة الإقلاع.**

ملخص:

- **airootfs/ = نظام الملفات الجذر الذي سيتم بناؤه داخل ملف الـISO.**
- **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.**

2.3ملفات الإعداد الأساسية(Core Configuration Files)

يمثل هذا الفصل حجر الزاوية في رحلتنا لتخصيص توزيعة آرتش 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 الأربعةprofiledef.sh — ، packages.x86_64، bootstrap_packages.x86_64، و — pacman.conf هو مفتاح تحقيق تحكم كامل بمشروعك، مما يمكنك من بناء توزيعة فريدة حقًا.

نحن لا نتحدث فقط عن قوائم الحزم؛ هذه الملفات تمثل "وصفة" متكاملة يتبعها archisoخطوة 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 توزيعتك 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 يتبعها archisoلإنتاج 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 shellcheckتجاهل تحذير معين يتعلق بالمتغيرات غير المستخدمة(SC2034) ، حيث قد تكون بعض المتغيرات مخصصة لأغراض محددة أو مستخدمة داخليًا بواسطة archiso.

بيانات التعريف الخاصة بـISO (ISO Metadata)

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) عادةً، يضيف archisoتلقائيًا 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>": معلومات الناشر. تتضمن اسم الجهة المسؤولة عن التوزيعة ونقطة اتصال (عنوان بريد إلكتروني في هذا المثال.)
- iso_application="Helwan Linux Live/Rescue DVD": وصف موجز لغرض الـ ISO. يوضح استخدامه المقصود، مثل "Live/Rescue DVD" أو."System Installer"
- iso_version="v1.1": رقم إصدار التوزيعة. هذا مهم لتتبع التحديثات وتحديد الإصدارات المحددة.

إعدادات البناء الأساسية(Core Build Settings)

تُعرّف هذه المتغيرات بنية وعناصر عملية البناء الأساسية.

- **install_dir="arch":دليل التثبيت داخل الـ .ISO عندما يقوم المستخدم بتحميل الـISO ، سيتم وضع ملفات النظام في هذا المسار. الافتراضي لـ Arch Linux هو .arch**
- **buildmodes=('iso'):أوضاع البناء المتاحة.**
  - **iso:يشير إلى أننا سنبني صورة ISO قابلة للإقلاع.**
  - **قد تشمل الأوضاع الأخرى (tarلإنشاء أرشيف لنظام الملفات) أو (vendorلإنشاء صور مخصصة لأجهزة معينة)، ولكن isoهو الأكثر شيوعًا.**
- **arch="x86_64":بنية المعالج المستهدفة. هنا، x86_64تعني أن الصورة مصممة لأنظمة 64 بت.**
- **pacman_conf="pacman.conf":اسم ملف إعدادات .Pacman يشير هذا إلى ملف pacman.confالموجود في نفس دليل ملف الإعدادات، والذي سيُستخدم 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 squashfsهو الأكثر استخدامًا لصور الـISO.**

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'):خيارات لأداة ضغط .squashfs**
  - **'xz' '-comp':يستخدم خوارزمية ضغط xz، والتي توفر نسبة ضغط عالية جدًا مقابل وقت معالجة أطول.**
  - **'x86' '-Xbcj':يحدد فلترًا خاصًا (x86 Code Branch) لتطبيق الضغط على شيفرة بنيةx86 ، مما يحسن نسبة الضغط.**
  - **'1M' '-b':حجم الكتلة (block size) الذي ستستخدمه الأداة. حجم أكبر يعني عادةً ضغطًا أفضل ولكنه يتطلب ذاكرة أكبر أثناء الضغط.**
  - **'1M' '-Xdict-size':حجم قاموس الضغط .(dictionary size) قاموس أكبر يؤدي بشكل عام إلى ضغط أفضل.**
- **bootstrap_tarball_compression=('zstd' '-c' '-T0' '--auto-threads=logical' '--long' '-19'):خيارات 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:--يُمكّن 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 الـISO.

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 (BIOSالقديم.)
  - 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:محمل إقلاع Syslinux.
  - eltorito:يستخدم معيار El Torito لإنشاء قرص CD/DVD قابل للإقلاع. هذا هو التنسيق القياسي لصور الـ ISOالقابلة للإقلاع.
- uefi-ia32.systemd-boot.esp:
  - uefi:يشير إلى وضع إقلاع UEFI (Unified Extensible Firmware Interface).
  - ia32:يشير إلى دعم المعالجات 32بت .(x86)
  - systemd-boot:يستخدم (systemd-bootالمعروف سابقًا بـ gummiboot)كمحمل إقلاع.UEFI
  - esp:يشير إلى هيكل الملفات داخل الـ ISO الذي يحاكي قسم نظام.(EFI (EFI System Partition - ESP

- **uefi-x64.systemd-boot.esp:**
  - **uefi:وضع UEFI.**
  - **x64:يدعم معالجات 64بت .(x86_64)**
  - **systemd-boot:محمل الإقلاع systemd-boot.**
  - **esp:هيكل ملفات ESP.**
- **uefi-ia32.systemd-boot.eltorito:**
  - **uefi:وضع UEFI.**
  - **ia32:يدعم 32بت.**
  - **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 64بت.**
  - **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 (airootfs) 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 ."owner:group:permissions"=["file_path"]

- owner:group: معرّف المستخدم (UID) ومعرّف 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 وقابلاً للقراءة بواسطة root فقط (400).
- /root: دليل المستخدم root. 750 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 سكربتات 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 السكربتات 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 التوزيعة (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 /airootfsوملفات الإقلاع.

2.3.2 packages.x86_64: قائمة حزم النظام الحي

إذا كان profiledef.shهو المخطط، فإن packages.x86_64هو قائمة المكونات الفعلية للنظام. هذا الملف هو قائمة نصية بسيطة بالحزم التي سيقوم archisoبتثبيتها داخل النظام الحي.(Live System)

فلسفة الاختيار

عند بناء توزيعة مخصصة، فكّر بعناية في الحزم التي تُدرجها. هناك توازن بين توفير بيئة غنية بالميزات والحفاظ على حجم ملف الـ ISOصغيرًا.

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 وlinux: هذه الحزم ضرورية لتوفير نظام قابل للإقلاع.**
- **أضف الأدوات الأساسية :أدوات مثل nano، vim، و gitغالبًا ما تكون ضرورية للمستخدمين.**
- **حدد بيئة سطح المكتب الخاصة بك :اختر بيئة سطح مكتب (مثلGNOME ، KDE Plasma، XFCE)أو مدير نوافذ (مثل i3، Sway)مع جميع الاعتماديات الضرورية.**
- **ضع في اعتبارك الأجهزة :إذا كانت توزيعتك تستهدف أجهزة معينة، فكّر في تضمين التعريفات الخاصة (مثل بطاقات Wi-Fi أو بطاقات الرسوميات.(**
- **إدارة الحزم :يستخدم archisoأداة pacman، ولكن يمكنك إضافة أدوات مساعدة مثل yayإذا 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_64ليست جزءًا من النظام الحي النهائي. بدلاً من ذلك، هي الحزم التي يحتاجها archisoنفسه 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 لتثبيتArch Linux ، وأهمها الأداة .arch-chrootتسمح هذه الأداة لـ archisoبالدخول إلى بيئة النظام الذي يتم بناؤه لإجراء عمليات التثبيت والتعديلات.**
- **archiso:هذه هي الأداة نفسها! يجب أن تكون متاحة داخل بيئة البناء.**
- **base:توفر الحد الأدنى من الحزم الضرورية لعمل نظامArch Linux ، بما في ذلك systemdوالأدوات الأساسية الأخرى.**
- **base-devel:تتضمن أدوات تطوير البرمجيات الأساسية (مثل gcc، makeالتي قد يحتاجها archisoلبناء مكونات معينة (على الرغم من أن هذا أقل شيوعًا في معظم عمليات بناء الـISO).**
- **linuxو linux-firmware:النواة (kernel) وبرامجها الثابتة المرتبطة بها، وهي ضرورية لبناء بيئة تشغيل أساسية.**

متى يتم تعديل هذا الملف:

في معظم السيناريوهات، لن تحتاج 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.confهو محور التحكم المركزي لـpacman ، وهو مدير الحزم الذي يشكل العمود الفقري لإدارة البرمجيات في Arch Linux.عندما تقوم ببناء صورة ISO باستخدامarchiso ، فإن هذا الملف يحدد كيفية جلب الحزم وتثبيتها، سواء داخل بيئة النظام الحي (live system) أو خلال مرحلة البناء الأولية .(bootstrap) دعنا نفصّل هذا الملف سطرًا بسطر، مع التركيز على الخيارات الأكثر أهمية لك كمطور توزيعة:

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 يخزّن فيها pacman قواعد بياناته(DBPath) ، حيث يحتفظ بالحزم التي تم تنزيلها(CacheDir) ، حيث يكتب سجلات العمليات(LogFile) ، وحيث يبحث عن مفاتيح (GPGDir) GPG 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 CacheDirموجودًا داخل مجلد مؤقت لمنع تلويث ذاكرة التخزين المؤقت للنظام المضيف.
- للتخصيص :نادرًا ما ستحتاج إلى تغيير هذه المسارات عند بناء صورةISO ، ولكن فهمها ضروري لفهم كيفية عمل pacmanداخليًا.

IgnorePkg / IgnoreGroup: وHoldPkg

```
HoldPkg = pacman glibc

IgnorePkg =

IgnoreGroup =
```

- :HoldPkgيمنع هذا الخيار الحزم المحددة من الترقية التلقائية. هذا أمر حاسم للحفاظ على استقرار النظام، خاصةً للحزم الأساسية مثل pacman و glibc نفسها.
- :IgnorePkgقائمة 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 تلقائيًا (مثل.(x86_64

- **في سياق archiso: بما أنك تبني صورة ISO لبنية محددة، يمكنك تعيينها صراحةً هنا (مثلاً Architecture = x86_64)، لكن auto تعمل جيدًا 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 لـISO ، مما يجعل هذا خيارًا مفيدًا 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 توقيعًا صالحًا (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 يدويًا) يمكن أن تكون غير موقعة أو موقعة بشكل اختياري.**
- **في سياق archiso: يضمن هذا الإعداد مستوى معينًا 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

- **[extra], [core]: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 افتراضيًا. 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 بتحديدها في packages.x86_64 و bootstrap_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 فريدًا.)**
- **SigLevel = Optional TrustedOnly:هنا، أنت تخبر pacman أنه يفضل أن تكون الحزم موقعة(Optional) ، ولكنه سيثق بها إذا كانت تأتي من مستودع موثوق به (TrustedOnly) حتى لو لم تكن التوقيعات الصريحة موجودة. يمكن أن يكون هذا مفيدًا للمستودعات المخصصة حيث قد لا تكون كل حزمة موقعة باستمرار.**
- **Server = https://helwan-linux.github.io/$repo/$arch:هذا هو عنوان URL للمستودع.**
  - **سيتم استبدال repo$تلقائيًا باسم المستودع (أي.(helwan**
  - **سيتم استبدال arch$بالبنية المستهدفة (مثلاً.(x86_64**

لماذا يهم؟ يتيح لك هذا تضمين حزمك المخصصة، أو حتى الإصدارات المعدلة من حزمArch ، مباشرةً 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 مفيدًا 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 محليًا داخل ملفات البروفايل، يمكنك استخدام هذا النوع من الإعداد.**

مثال عملي توضيحي

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

1. **أنشئ مستودعك المخصص:**
   - **أنشئ دليلًا لمستودعك، مثلاً ~/my_helwan_repo.**
   - **ضع ملف حزمتك (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 محليًا.**
3. **عدّل pacman.conf: إذا قمت برفعه إلى https://helwan-linux.github.io/helwan/x86_64/، 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 حزمتك المخصصة إلى القائمة:**
8. **... #حزم أخرى...**

9. helwan-tools

10. ... #حزم إضافية...

الآن، عندما يقوم archiso ببناء صورةISO ، سيعرف pacman من أين يجلب helwan-toolsوسيقوم بتثبيته في النظام الحي.

شرح إضافي(profiledef.shمن) pacman.conf

لقد قدمت أيضًا مقطعًا مثالًا من ملف profiledef.shيوضح استخدام pacman.confكملف خارجي:

Bash

"pacman.conf"=pacman_conf يشير إلى ملف pacman.conf الموجود في نفس دليل البروفايل

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

ملخص الفصل

ملف pacman.confهو أداة قوية لتخصيص مصادر الحزم الخاصة بك والتحكم في كيفية جلب الحزم وتثبيتها. 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 /airootfsبعد 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 /airootfsكبيئة لنظام الملفات الجذر.(root filesystem)**
- **شرح الدلائل الفرعية:**
  - **:/etcإعدادات النظام.**
  - **:/rootدليل المستخدم الجذر.**
  - **:/usrملفات ثنائية إضافية.**

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

:xorg.conf.d/30-touchpad.conf * X11 يضبط إعدادات لوحة اللمس (touchpad) لبيئة Xorg الرسومية، مثل حساسية اللمس والتمرير :grub * default . ملف إعداداتGRUB ، 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 * lightdmالإعدادات الرئيسية لـLightDM ، وهو مدير عرض (display manager) للواجهة الرسومية.

- **:slick-greeter.confإعدادات شاشة تسجيل الدخول (greeter) لـ :* locale.conf . LightDM يحدد إعدادات اللغة والمنطقة للنظام، مثل تنسيق التاريخ والوقت والأرقام :* localtime . يشير إلى الوقت المحلي الذي تم إعداده للنظام . :* mkinitcpio.confملف إعدادات لـ mkinitcpio، يُستخدم لإنشاء صور) initramfsنظام 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 . initramfsيحدد 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 حزمArch Linux ، يحدد المستودعات (repositories) وإعدادات التحديث :hooks/uncomment-mirrors.hook * pacman.d . خطاف (hook) لـ Pacmanيقوم بإلغاء التعليق عن خيارات المرآة (mirror) لإعدادات التحديث.**

- **:hooks/zzzz99-remove-custom-hooks-from-airootfs.hookخطاف آخر لـ Pacmanمصمم لإزالة الخطافات المخصصة من :* passwd . airootfsيحتوي على معلومات المستخدم الأساسية، مثل أسماء المستخدمين، UIDs، والدلائل الرئيسية :rules.d/49-nopasswd_global.rules * polkit-1 . (home directories) 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, .Xresourcesملفات إعدادات شائعة لبيئة) X مثل إعدادات المظهر) و) shell Bash الأوامر والأسماء المستعارة.(**

- **cinnamon:.إعدادات وتهيئة خاصة لبيئة سطح المكتبCinnamon ، 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، السكربتات-sshd_config.d/10 * ssh). archiso.conf:إعدادات إضافية لخادم SSH خاصة ببنية :g_wheel * sudoers.d .Archiso ملف إعدادات لـ sudo، يمنح أعضاء مجموعة wheelصلاحيات 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 الشبكةEthernet) ، WLAN، .(WWAN**
- **network.conf.d/ipv6-privacy-extensions.conf:تمكين امتدادات الخصوصية لـ.IPv6**
- **resolved.conf.d/archiso.conf:إعدادات خاصة لـ) systemd-resolvedنظام حل الأسماء) لـ.Archiso**
- **system:يحتوي على ملفات service.و target.مختلفة تحدد كيفية تشغيل الخدمات وإقلاع النظام. أمثلة تشمل) display-manager.serviceلبدء الواجهة الرسومية(،) NetworkManager.serviceلإدارة الشبكة(،) sshd.serviceلخادم(SSH ، و) reflector.serviceلتحسين 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.**

2.4.1.2دليل 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 لتشغيل سلسلة من الأوامر تلقائيًا، مما يوفر الوقت والجهد للمستخدم الجذر عند أداء الإجراءات المتكررة. يمكن أن تتضمن أوامر لتثبيت 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) GnuPG أداة لتشفير وفك تشفير البيانات والتوقيع الرقمي. النقطة (.) في البداية تجعل الدليل مخفيًا.**
  - **التقنية:**
    - **scdaemon.conf:هذا هو ملف إعدادات لـ(scdaemon (Smartcard Daemon ، وهو مكون من GnuPG يتعامل مع البطاقات الذكية (smart cards) والأجهزة التشفيرية الأخرى. يحدد هذا الملف كيفية تفاعل scdaemon 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 (Z 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.).

usr/
 usr/
├── local
│ ├── bin
│ │ ├── Installation_guide
│ │ ├── choose-mirror
│ │ ├── launch
│ │ └── livecd-sound
│ ├── share
│ │ ├── livecd-sound
│ │ └── asound.conf.in
└── share
├── backgrounds
│ ├── bg.png
│ ├── helwan
├── grub
│ ├── themes
│ ├── helwan
├── nemo
└── actions

└── count_lines.nemo_action

2.4.1.3فهم بنية نظام الملفات - دليل usr/ بعد استيعاب المفاهيم الأساسية للملفات الضرورية ودور دليل aiirootfsكبيئة 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 والمشتركة 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، صفحاتman ، والتوثيق الآخر.**

نظرة تفصيلية على محتويات usr/ في سياق Archiso في سياقArchiso ، الأداة المستخدمة لإنشاء صور ISO قابلة للإقلاع من Arch Linux، يعكس دليل usr/داخل aiirootfs بنية 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) لـGRUB ، مما يسمح بتخصيص شاشة الإقلاع.
- helwan: دليل فرعي يمثل سمة مخصصة لـGRUB ، 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 إجراءً 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 والسكريبتات 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 وloopback.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 فهمًا عميقًا للمكونات الداخلية لعملية الإقلاع، مما يتيح لك التعامل مع بيئات GRUB بمرونة وثقة.

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

شرح مفصل لملفgrub.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 وحداتGRUB**

```
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_gptو 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 الملفاتFAT16/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": يتم 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_input --append serial: و terminal_output --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 "${ARCHISO_UUID}"]: يتحقق هذا مما إذا كان متغير ARCHISO_UUID فارغًا، مما يعني أنه لم يتم التعرف على قرص Archiso بعد.
- if [-z "${ARCHISO_HINT}"]: يتحقق هذا مما إذا كان لا يوجد تلميح (hint) موجود مسبقًا.
- regexp --set=1:ARCHISO_HINT '^\(([^)]+)\)' "${cmdpath}": يحاول هذا استخراج تلميح (مثل مسار أو اسم ملف) من متغير cmdpath، الذي يمثل المسار الذي تم استدعاء GRUB منه.
- search --no-floppy --set=root --file '%ARCHISO_SEARCH_FILENAME%' --hint "${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}": يفحص هذا الجهاز المحدد بواسطة ${root} للحصول على معرف UUID (Universally Unique Identifier) 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 بالثواني 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 }
```

menuentry "..." --class ... --id 'archlinux': يبدأ هذا الأمر تعريف إدخال في القائمة بالاسم "Helwan installer(x86_64, UEFI)". تُستخدم class-- و id--لتصنيف الإدخال وتحديده 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)

- /%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) مبكرًا.

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

- intel-ucode.img و amd-ucode.img: ملفات تحديث الميكروكود (microcode) لمعالجات Intel وAMD ، لتحسين الاستقرار والأداء.
- initramfs-linux.img: ملف initramfs الرئيسي الذي يحتوي على الأدوات اللازمة لبدء تشغيل النظام.

7.2.إدخالات خاصة بـ UEFI يظهر هذا القسم فقط إذا كان GRUB يعمل في وضع UEFI.

if ["${grub_platform}" == "efi"]; then: يتحقق مما إذا كان GRUB يعمل على نظام UEFI. if ["${grub_cpu}" == "x86_64"]; then: إذا كان معالج النظام 64 بت ... menuentry "Run Memtest86+ (RAM test)" * :إدخال لتشغيل Memtest86+ ، أداة لاختبار ذاكرة الوصول العشوائي: /boot/memtest86+/memtest.efi * linux (RAM). يقوم بتشغيل تطبيق Memtest86+ بتنسيق EFI. * menuentry ... "UEFI Shell": إدخال لتشغيل UEFI Shell ، وهي واجهة سطر أوامر يوفرها بيئة UEFI. * insmod chain: يقوم بتحميل الوحدة النمطية chain، التي تسمح بـ "سلسلة" تحميل برنامج آخر * chainloader /shellx64.efi: .يقوم بتحميل وتشغيل تطبيق UEFI Shell 64 بت elif ["${grub_cpu}" == "i386"]; then: 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.إدخالات 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 (الرسومية والمتسلسلة)، والبحث عن بيئة Archiso ، ثم يعرض قائمة بخيارات الإقلاع بما في ذلك مثبت Arch Linux ، والأدوات المفيدة مثل Memtest86+ وUEFI Shell ، بالإضافة إلى خيارات إيقاف التشغيل وإعادة التشغيل.

شرح ملف loopback.cfg لـ GRUB: 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 :"${iso_path}" --file --set=archiso_img_dev --no-floppy search ISO يبحث هذا الأمر عن ملف صورة ISO المحدد في المتغير .${iso_path}الخيار no-floppy--يخبر GRUB بتجاهل محركات الأقراص المرنة، و --set=archiso_img_devيعين متغيرًا داخليًا يسمى archiso_img_devللإشارة إلى الجهاز الذي تم العثور عليه --set probe . :"${archiso_img_dev}" --fs-uuid archiso_img_dev_uuidبعد العثور على الجهاز، يقوم هذا الأمر بفحصه للحصول على معرف نظام الملفات الفريد (UUID) الخاص به ويخزنه في المتغير .archiso_img_dev_uuidهذا ضروري لضمان أن GRUB يمكنه دائمًا تحديد موقع قسم ISO بشكل موثوق، حتى لو تغير ترتيب الأقراص.

تحديد معرف المنصة (Platform Identifier) الأوامر التي تبدأ بـ [... == "${grub_platform}"] ifتبني متغير archiso_platformلوصف بيئة الإقلاع الحالية بدقة.

- إذا كان grub_platformهو'efi' (UEFI) ، فإنه يتحقق من بنية المعالج (grub_cpu) ويضيف 'x64' لـ (x86_64 أو 'IA32'لـ (i386 أو ببساطة اسم المعالج إذا كان مختلفًا.
- إذا كان grub_platformهو'pc' ، يتم تحديد المنصة على أنها.'BIOS'
- في الحالات الأخرى، يتم استخدام مزيج من grub_cpuو grub_platformلإنشاء معرف فريد. هذا يضمن عرض خيارات الإقلاع الصحيحة بناءً على ما إذا كان النظام يستخدم UEFI أوBIOS ، وبنية المعالج.

إعدادات القائمة الافتراضية

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

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

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

تحديد خيارات القائمة

كل قسم يبدأ بـ { ... } "..." menuentryيمثل خيارًا واحدًا في قائمة.GRUB

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

- :class arch --class gnu-linux --class gnu --class osتُستخدم هذه الفئات لتصنيف القائمة وتنسيقها، وغالبًا ما تُستخدم في واجهات GRUB الرسومية.
- :'id 'archlinux--معرّف فريد لهذا الخيار، يُستخدم في سطر .default=archlinux
- :set gfxpayload=keepيحافظ على وضع الرسوميات الحالي إذا كان موجودًا.
- :linux /%INSTALL_DIR%/boot/%ARCH%/vmlinuz-linux ...هذا الأمر يحمّل 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 المعرّف الفريد للجهاز الذي يحتوي على صورة الـ ISOإلى النواة.

- img_loop="${iso_path}": يمرر مسار ملف الـ ISO إلى النواة.

- initrd /%INSTALL_DIR%/boot/%ARCH%/initramfs-linux.img: هذا يحمّل 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 مناسبًا 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 مثالًا عمليًا 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 سلسة عند الإقلاع من وسائط قابلة للإزالة.

ملخص تقني احترافي

- المرونة: يدعم الملف الإقلاع عبر كل من UEFI وBIOS ، بالإضافة إلى المعالجات 64 بت و 32 بت.
- الكشف التلقائي: يكتشف تلقائيًا موقع صورة الـ ISO ومعرّفها الفريد، مما يجعله مرنًا 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 إدخالات 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 دائمًا 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 خصيصًا لـ Archiso. وظيفتها الرئيسية هي فحص بيئة الإقلاع الحالية لتحديد ما إذا كان الإقلاع من وسائط محلية أو عبر الشبكة (PXE).**
- **APPEND -pxe- pxe -sys- sys -iso- sys: يمرر هذا الأمر معاملات إلى الوحدة whichsys.c32. تُستخدم هذه المعاملات لتوجيه الوحدة إلى "منصات" محددة.**
  - **-pxe- pxe-: إذا تم اكتشاف إقلاع PXE، فسيتم إعادة التوجيه إلى تسمية pxe.**
  - **-sys- sys-: إذا تم اكتشاف الإقلاع من وسائط محلية (مثل محرك أقراص USB)، فسيتم إعادة التوجيه إلى تسمية sys.**
  - **-iso- sys-: يضمن هذا المعامل أن الإقلاع من صورة ISO يذهب أيضًا 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 تلقائيًا، 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 مُحمّل 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 بصريًا، 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 لمُحمّل الإقلاع.**

أبعاد وتخطيط القائمة (Menu Dimensions and Layout)

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

- **MENU WIDTH 78: يضبط عرض نافذة القائمة الرئيسية على 78 حرفًا.**
- **MENU MARGIN 4: يضبط الهامش الأيسر للقائمة على 4 أحرف.**
- **MENU ROWS 7: يحدد عدد صفوف القائمة المرئية.**
- **MENU VSHIFT 10: يقوم بإزاحة القائمة رأسيًا بمقدار 10 صفوف.**
- **MENU TABMSGROW 14, MENU CMDLINEROW 14, MENU HELPMSGROW 16, MENU HELPMSGENDROW 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 وخلط 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 عرضًا نظيفًا.**
- **MENU IMMEDIATE: يتسبب هذا الأمر في عرض القائمة فورًا دون انتظار ضغطة مفتاح من المستخدم.**

باختصار، ملف archiso_head.cfg ليس ملفًا قابلاً للإقلاع بحد ذاته. إنه مكون أساسي يتم تضمينه بواسطة ملفات الإعدادات الأخرى لإنشاء بيئة إقلاع متسقة ومصممة جيدًا 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 طرقًا 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 أيضًا 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 مُحمِّل 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.**
  - **MENU LABEL Run Memtest86+ (RAM test): 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 المعلمات ملفات التكوين إلى HDT. يساعد modules_alias.gz HDT على ربط أجهزة الأجهزة بوحدات النواة الصحيحة، بينما يوفر pciids.gz خريطة لمعرفات 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.**

بشكل أساسي، يقوم archiso_tail.cfg بدمج أدوات النظام الأساسية في قائمة سهلة الاستخدام، مما يكمل خيارات تثبيت نظام التشغيل الأساسي أو البيئة 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 جزءًا 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 سابقًا، 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 إدخالات القائمة الفعلية للإقلاع عبر PXE. هنا، يتم تحديد المسارات إلى نواة Linux وصورة initramfs التي سيتم تحميلها عبر الشبكة، جنبًا 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 إدخالات 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 صيانته.**
  - **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% cms_verify=y archiso_nbd_srv=${pxeserver}: 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 صيانته.**
  - **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 معلمات 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 صيانته.**
  - **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}/ cms_verify=y: يمرر معلمات النواة لتحديد خادم HTTP.**

فهم archiso_sys-linux.cfg للإقلاع عبرBIOS

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

فيما يلي تفصيل لكل مكون:

- **LABEL arch64:هذا هو مُعرّف فريد ل entryالإقلاع المحدد هذا. بهذه الطريقة يشير مُحمِّل 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).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.imgو amd-ucode.imgهي تحديثات دقيقة للمعالجات من Intel و AMD على التوالي. تساعد هذه في ضمان التهيئة الصحيحة للأجهزة واستقرارها.**
  - **initramfs-linux.imgهو 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:يمرر 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 الرسوميات 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: يضمن تحميل تعريفات وحدات تخزين NVMe ، وهو أمر بالغ الأهمية للأنظمة التي تستخدم أقراص NVMe SSD.**

بشكل 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 والمعلمات الضرورية بشكل صحيح لعملية إقلاع وتثبيت ناجحة.

2.6 أسرار الدليل /etc/

مقدمة

عندما نتحدث عن بناء توزيعة 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 تخدش السطح فقط. الجوهر، الشخصية الحقيقية للتوزيعة، 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 التوزيعة:

- كيف يتعامل النظام مع المستخدمين وصلاحياتهم.
- كيف تتم إدارة الشبكات والخدمات.
- لغة التوزيعة، 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 يونكس 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 يونكس ثم جنو/لينكس، تحول تدريجيًا إلى مركز ثقل النظام، حتى أصبح اليوم بمثابة "دستور" التوزيعة.

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 توزيعة 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)

├── systemd والخدمات (network, display, cloud-init, zram …)

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

- يمثل passwdو shadowالعمود الفقري لإدارة الحسابات.
- يحدد pacman.confبدقة من أين يأتي البرنامج وكيف يتم التحقق من سلامته.
- يقوم lightdm.confو slick-greeter.confبتكوين الشاشة الأولى التي يراها المستخدم.
- تضمن خدمة reflector.serviceتحديث المرايا تلقائيًا للحصول على تثبيتات وتحديثات سريعة ومستقرة.
- يعكس الدليل skel/Templates/hel-filesالبصمة 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 توزيعة. 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.**
- **systemdوالخدمات.**
- **إعدادات المستخدم الرسومية.**

سنركز على أكثر من ستين ملفًا ودليلًا، 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 بسيطًا، هو خيط في نسيج كامل.

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

2.6.2إدارة المستخدم والأمان

ملف /etc/passwd

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

- **root:x:0:0:root:/root:/usr/bin/zsh**
- **liveuser:x:1000:1000::/home/liveuser:/bin/bash**
- **:rootالمستخدم ذو أعلى الامتيازات (المشرف الخارق.(**
- **:UID = 0صلاحيات مطلقة.**
- **shell = zshبدلاً من :bashقرار تصميمي في Helwan Linux لتجربة أكثر حداثة.**
- **:liveuserالمستخدم لجلسة .Live**

- :1000 = UIDأول مستخدم عادي.
- :bash = shellبسيط وسهل للمبتدئين.

الفلسفة :فيUnix ، "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 Liveدون 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
- :wheelللوصول إلى .sudo
- :networkلإدارة الشبكة.
- :audio/video/storageلتشغيل 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) wheelمثل (liveuserتنفيذ أوامر sudoدون كلمة مرور .هذا مفيد لبيئة Liveولكنه عادةً ما يتم تغييره بعد التثبيت 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 التثبيت.

/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 التثبيت العالمي ويسمح بتخصيص المنطقة الزمنية في مرحلة لاحقة.

2.6.4. إدارة الإقلاع والتهيئة

/etc/default/grub

يوفر هذا الملف تحكمًا 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 التدقيق لتسريع عملية الإقلاع.**
- **GRUB_THEME="/usr/share/grub/themes/helwan/theme.txt": يطبق سمة مخصصة لمظهر جذاب واحترافي.**
- **GRUB_DISABLE_OS_PROBER=false: يُمكّن 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 متخصصة 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 COMPRESSION_OPTIONS=(-9e): يستخدم ضغطًا عاليًا جدًا 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 فورًا 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 أيضًا تحديثات النظام، وإزالة الحزم غير المرغوب فيها، والتحقق من سلامة الحزم من خلال التوقيعات الرقمية.

4.1 pacman.conf — 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 تلقائيًا.
  - 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 لتوزيعة Helwan Linux ، والذي يتضمن حزمًا مخصصة تم ضبطها خصيصًا للتوزيعة.
  - [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:

ملاحظة :يمكن إضافة مستودعات أخرى مثل multilibأو مخصصة، ولكنها معطلة افتراضيًا.

4.2خطافات — pacmanالأتمتة

بالإضافة إلى تكوين pacman.conf، يمكن التحكم في سلوك pacmanبعد العمليات باستخدام الخطافات .(hooks) يحتوي الدليل /etc/pacman.d/hooks/على نصوص برمجية صغيرة يتم تنفيذها تلقائيًا 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-mirrorlistأو تحديثها.
  - الإجراء :(Action) يستخدم أمر sedلإلغاء التعليق على جميع الخوادم في /etc/pacman.d/mirrorlist، مما يؤدي فعليًا إلى تمكين جميع المرايا تلقائيًا.
  - 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 نصًا برمجيًا 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 pacmanفي بيئة ISO دون التأثير على النظام بعد التثبيت.

4.3خاتمة

- يحدد pacman.confسياسات) pacmanالأمان، المستودعات، خيارات التنزيل.(
- توفر الخطافات الأتمتة لعمليات ما بعد التثبيت أو الترقية.
- دمج مستودع Helwan Linux المخصص يجعل التوزيعة مكتفية ذاتيًا، ومتميزة عنArch ، مع الاحتفاظ بمتانة النظام الأساسي.

2.6.6إدارة المرايا(Mirrors)

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

2.6.6.1ملف تكوين reflector

الموقعetc/xdg/reflector/reflector.conf/ :

يحدد هذا الملف كيفية عمل أداة reflectorعند 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 mirrorlistباستمرار بأفضل الخوادم، مما يسرع تنزيل الحزم بشكل كبير.

2.6.6.2خدمة reflectorفي systemd

الموقعetc/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 reflectorباستخدام ملف التكوين المذكور أعلاه.
- تحديث etc/pacman.d/mirrorlist/تلقائيًا.
- ضمان أن pacmanيستخدم 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.

2.6.6.3تكامل خدمات pacmanوخطافاتها(Hooks)

لا يعتمد Helwan Linux فقط على pacman.confوالمرايا، ولكنه يستخدم أيضًا مزيجًا من خدمات systemdوخطافات pacman.

- **خطافات) pacmanكما رأينا سابقًا) تنظف وتعدل التكوينات أثناء تثبيت الحزم أو ترقيتها أو إزالتها.**
- **خدمة reflectorتقوم بتحديث المرايا تلقائيًا.**

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

2.6.6.4الخلاصة

- **pacman.conf:مركز التحكم لمدير الحزم.**
- **خطافات :pacmanأتمتة ذكية أثناء التثبيت/الترقيات.**
- **reflector:اختيار تلقائي لأفضل الخوادم.**
- **خدمات :systemdتضمن تحديث المرايا المستمر دون تدخل المستخدم.**

يجمع Helwan Linux كل هذه العناصر لتقديم نسخة جاهزة للاستخدام منArch Linux ، مما يلغي التكوين اليدوي الشاق الذي غالبًا ما يردع معظم المستخدمين.

2.6.7 pacman-init.service:تهيئة pacmanعند أول إقلاع

الموقعpacman-init.service/etc/systemd/system/ :

2.6.7.1الغرض من الخدمة

يعتمد مدير الحزم pacmanعلى 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 (مُعبئي الحزم الرسميين في + Arch مستودع.(Helwan**

2.6.7.2شرح 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 (لصلاحية 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 سلسة، 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 توزيعة 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، سيقرأ pacman دائمًا من قائمة الخوادم (mirrorlist) التي تم إنشاؤها تلقائيًا. هذا يوفر للمستخدمين سرعة في تثبيت وتحديث الحزم من أقرب وأسرع الخوادم.**
- **--ipv4 و --ipv6: تتيح هذان الخياران لـ reflector جلب الخوادم التي تدعم كلا البروتوكولين. هذا أمر بالغ الأهمية لكي تعمل التوزيعة 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 تلقائيًا.**
- **--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 التوزيعة: **

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

- يوفر إعدادات افتراضية ذكية: يمنح Helwan Linux تكوينًا ذكيًا من البداية، بدلاً من إجبار المستخدم على البحث يدويًا عن الخوادم.

LightDM: 2.6.9مدير العرض الرسومي LightDMهو المكوّن المسؤول عن شاشة تسجيل الدخول في توزيعة Helwan Linux. ملف التكوين الرئيسي الخاص به يوجد في المسار /etc/lightdm/lightdm.conf :هذا هو الملف الأساسي الذي يتحكم في كيفية عملLightDM ، 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 فيlightdm.conf [LightDM]

- run-directory=/run/lightdm:يحدد المسار الذي يستخدمه LightDM لتخزين بيانات التشغيل (ملفاتPID ، مآخذ التوصيل.(
- log-directoryو) 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:نص وسيط يستخدمه LightDM لإطلاق جلسة المستخدم .تشمل الإعدادات الأخرى:
- autologin-user:يُمكّن 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 [XDMCPServer] .هذا القسم يسمح بالاتصالات عن بعد باستخدام بروتوكول .XDMCP وهو معطل افتراضياً (enabled=false)لأسباب أمنية [VNCServer] .يُمكّن هذا القسم تسجيل الدخول عن بعد عبر .VNC وهو معطل أيضاً افتراضياً. يمكن أن يكون مفيداً في بيئات الدعم الفني أو الفصول الدراسية.

slick-greeter: 2.6.9.1تخصيص شاشة تسجيل الدخول ملف التكوين الخاص بـ slick-greeter يوجد في المسار : /etc/lightdm/slick-greeter.confهذا 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=Qogirو :cursor-theme-size=16 مظهر وحجم مؤشر الماوس.

- draw-user-backgrounds=false: يمنع استخدام خلفية شخصية لكل مستخدم.
- show-a11y=false و show-power=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 إدخالات 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 DHCP ISO 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 بسيطًا في البيئات الحية، حيث تتعامل أدوات مثل NetworkManager أو iwd مع اتصالات Wi-Fi الفعلية. تضمن 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 IPv6 (RFC 4941) 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.

ipv6-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: يمكّن SSH خصيصًا لبيئة النظام الحي؛ ويجب تحذير المستخدمين من تركه مفعّلًا على نظام مثبت.

livecd-alsa-unmuter.service

الهدف :خدمة مصممة لإزالة كتم صوت أجهزة ALSA افتراضيًا 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 سلسة: 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 :يمكّن 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.**
- **agetty --autologin liveuser:يسجل دخول liveuserمباشرةً دون الحاجة لكلمة مرور.**
- **noclear:--يمنع مسح شاشة الطرفية بعد تسجيل الدخول.**

ملاحظات :Helwanهذه ميزة ضرورية لسهولة الاستخدام، خاصةً 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. فيHelwan 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 gettyمن العمل بالتزامن مع LightDM.**
- **ExecStart=/usr/bin/lightdm:هنا، كان الاختيار هو) LightDMمع slick-greeter).**
- **WantedBy=graphical.target:يضمن بدء LightDMعند الدخول إلى الوضع الرسومي.**

ملاحظات :Helwanربط الخدمة بـ LightDMيضيف تجربة مستخدم رسومية مباشرة. بعد التثبيت، يمكن للمستخدمين التغيير إلى أي مدير عرض آخرGDM) ، .(...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/reflector --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=reflector ...:يستخدم أداة Reflectorلتحديد أسرع 20 خادمًاHTTPS ، ويرتبها حسب سرعة الاتصال.**
- **WantedBy=multi-user.target:يضمن بدء هذه الخدمة تلقائيًا عند الدخول إلى وضع المستخدمين المتعددين (multi-user mode).**

ملاحظات :Helwanهذه الخطوة عبقرية لأنها تقلل بشكل كبير من وقت التثبيت وتوفر تجربة سلسة، حتى للمستخدمين البعيدين جغرافيًا عن الخوادم الأساسية لـArch Linux.

reflector.service

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

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 يدويًا أو ربطها بخدمة مؤقت (timer) للتحديثات المجدولة.**

ملاحظات :Helwan هذا يضمن أن النظام لا يظل مرتبطًا 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 تلقائيًا مرة واحدة كل أسبوع.**
- **WantedBy=timers.target: يسجل هذا المؤقت ضمن نظام إدارة المؤقتات في systemd.**

ملاحظات :Helwan هذا حل ذكي يضمن تحديث المستودعات دون تدخل يدوي. يمكن للمستخدم نسيان الأمر تمامًا، 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.

/etc/systemd/system/graphical.target

الغرض:
هذا ملف هدف (target) وليس خدمة. يمثل حالة التشغيل للنظام في وضع واجهة المستخدم الرسومية.(GUI)

المحتوى النموذجي:

```
[Unit]

Description=Graphical Interface

Requires=multi-user.target

After=multi-user.target

AllowIsolate=yes
```

الشرح:

- Requires=multi-user.target: يجب أن يبدأ نظام الجلسة النصية (CLI) قبل واجهة المستخدم الرسومية.
- After=multi-user.target: لضمان الترتيب الصحيح لتسلسل تشغيل الخدمات.
- AllowIsolate=yes: يسمح بالتبديل يدويًا إلى وضع الواجهة الرسومية باستخدام:
- systemctl isolate graphical.target

ملخص مكونات systemd:

- choose-mirror.service: يختار أسرع المرايا أثناء الإقلاع.
- reflector.service + reflector.timer: يقوم بتحديث المرايا بشكل دوري (أسبوعيًا).
- systemd-networkd.service: يدير الشبكات بطريقة خفيفة.
- systemd-resolved.service: يحل أسماء النطاقات ويخزنها مؤقتًا.
- systemd-timesyncd.service: يزامن الوقت مع خوادم NTP.

- dhcpcd.service: يحصل على عنوان IP تلقائيًا عبر DHCP.
- getty@tty1.service: يوفر واجهة تسجيل دخول نصية كنسخة احتياطية.
- graphical.target: يمثل وضع واجهة المستخدم الرسومية (GUI).

2.6.11 وظيفة دليل /etc/skel

دليل /etc/skel، المعروف أيضًا باسم "دليل الهيكل العظمي"، يعمل كقالب للحسابات الجديدة للمستخدمين. عند إنشاء مستخدم جديد باستخدام الأمر useradd مع الخيار (m-الذي يضمن إنشاء دليل المنزل)، يتم نسخ محتويات /etc/skel/ تلقائيًا إلى دليل المنزل الخاص بالمستخدم الجديد .(/home/username/) أي ملف أو مجلد موجود في /etc/skel/ سيتم تكراره لكل مستخدم جديد.

الاستخدامات الأساسية:

- تخصيص بيئة المستخدم الافتراضية :يشمل ملفات إعدادات الشل الأساسية مثل .bashrc، .zshrc، .profile، و.bash_profile، التي تسمح بضبط الإعدادات المسبقة للشيل، الاختصارات، ألوان سطر الأوامر، وتفضيلات المستخدم الأخرى.
- إعداد بيئة سطح المكتب :يمكن وضع ملفات ضمن /.config لإعداد بيئة سطح المكتب مسبقًا، بما في ذلك إعدادات محررات النصوص، السمات، الخلفيات، أو ملفات القوالب للمستخدمين الجدد.
- توفير تعليمات أو مستندات للمستخدم الجديد :يمكن تضمين ملفات ترحيبية مثل README أو WELCOME، أو ملفات نصية تحتوي على تعليمات البدء أو روابط مفيدة.

التطبيق في توزيعات مبنية على) Arch مثل:(Helwan Linux

في توزيعات مثلHelwan Linux ، قد يكون دليل /etc/skel/ فارغًا أو يحتوي على ملفات أساسية فقط (مثل .bashrc الافتراضي). لكن كمطور توزيعة يمكنك استخدام هذا الدليل لـ:

- ضبط إعدادات الشل أو سطح المكتب مسبقًا :إضافة إعدادات جاهزة للاستخدام للمستخدم الجديد.
- إنشاء تجربة مستخدم أولية متسقة :ضمان مظهر ووظائف موحدة لكل المستخدمين الجدد.
- توفير بيئة معدة مسبقًا :السماح لكل مستخدم بالبدء ببيئة مجهزة بدلًا من بيئة فارغة.

نص مقترح للتوثيق:

— /etc/skel/ 2.xالقالب الأولي للمستخدمين الجدد

يعمل دليل /etc/skel/كقالب افتراضي يُنسخ تلقائيًا إلى دليل المنزل لكل مستخدم جديد يتم إنشاؤه باستخدام useradd -m. أي ملف أو مجلد ضمن /etc/skel/ سيتم تكراره في /home/username/.
تُستخدم ملفات الإعداد مثل .bashrc و.profile لتخصيص بيئة الشل. يمكن إضافة إعدادات مخصصة في /.config لتحضير بيئة سطح المكتب أو محررات النصوص، وكذلك تضمين ملفات ترحيبية أو تعليمات للمستخدم الجديد.

في توزيعات مثلHelwan Linux ، يعد هذا الدليل نقطة رئيسية لتوحيد تجربة المستخدم الافتراضية، وضمان أن جميع المستخدمين يبدأون بإعدادات معدة مسبقًا بدلًا من بيئة خام.

2.7في المختبر(In the Lab)

في هذا القسم من الكتاب، سنتعلم عمليًا كيفية تخصيص توزيعة بطريقة متماسكة وسليمة .سنبدأ من الصفر للاستفادة مما ناقشناه في الأقسام السابقة.

مهما كانت ميولك أو اتجاهاتك، لا تتردد في وضع بصمتك الخاصة على التوزيعة .اجعل التوزيعة تُعلن هويتها بصوت عالٍ. هذا هو ما سيجعل توزيعتك فريدة. لا تحاول إرضاء المستخدمين على حساب اتجاه التوزيعة؛ محاولة إرضاء الجميع ستؤدي إلى فقدان الجميع، وتحويل مشروعك إلى نسخة فوضوية بلا هوية، أو مجرد تقليد لـ "موناليزا"، أو إعلان صاخب لمحاولة مهرج تقليد ممثل مشهور.

اختيار الحزم بعناية :

تحديد الحزم هو ما يمنح التوزيعة طعمها وخصائصها، وليس مجرد شكلها ومظهرها. لذلك، لا تُرهق إدارة الحزم الخاصة بك بسهولة ولا تجعلها مصدر فشل بدلًا من النجاح. اختر الحزم بعناية قبل إضافتها للتوزيعة. اسأل نفسك:

- لماذا هذه الحزمة بالذات؟
- هل ستعزز هوية التوزيعة؟
- هل ستخدم المستخدم المستهدف؟
- هل يمكنني برمجة حزمة مشابهة توفر على المستخدم صعوبة استخدام هذه الحزمة؟

لا تتسرع في حشو التوزيعة بالحزم؛ كل حزمة مُضافة يجب أن تكون مدروسة جيدًا وتخدم غرض التوزيعة.

ملف – profiledef.shصديقك الوفي

بطاقة مهمة أخرى تبرز في اللعبة هي ملف .profiledef.shدور هذا الملف لا يقتصر على تحديد اسم التوزيعة فقط، بل يشمل أيضًا إدارة أذونات الملفات.(file_permissions)

يمكنك اعتباره صديقك الوفي، الذي يسهل عليك إنجاز المهام دون أن تتحمل عبء كل شيء بنفسك. يعمل هذا الملف بثبات في الخلفية، وينجز المهام القوية بسلاسة، مثل موظف مجتهد لا يكل أبدًا.

هل أنت مستعد للدخول إلى المختبر لتفكيك نواة التوزيعة بعناية؟

تثبيتArchiso

يُعد Archisoالأداة الرسمية لإنشاء صور ISO قابلة للإقلاع للتوزيعات المستندة إلىArch.
لتثبيته علىArch Linux ، استخدم الأمر التالي:

```
sudo pacman -S archiso
```

هذا يكفي لإعداد بيئة العمل الخاصة بك لإنشاء الـ ISO. يُنصح أيضًا بتثبيت الأدوات المساعدة مثل base-devel، وvim، و gitإذا كنت تخطط لتخصيص التوزيعة بشكل واسع.

2.فهم اختلافات الملفات الشخصية(Profiles)

بعد تثبيتarchiso ، ستجد الملفات الشخصية المتاحة في:

/usr/share/archiso/configs/

الملفات الشخصية الرئيسية:

Profile	الوصف
releng/	الملف الشخصي الرسمي للإصدار. يمثل نسخة Arch Linux المستقرة الحالية ويستخدم لإنشاء ISO مطابق للتوزيعة الأساسية.
baseline/	نسخة مختصرة من Arch تحتوي على الحزم الأساسية فقط لتقليل حجم ISO.
next/	ملف تجريبي/تطويري يشمل أدوات جديدة واختبارات لم تعتمد رسميًا بعد.

ملاحظات عملية:

- **معظم مشاريع التوزيعات الجديدة تبدأ بـ releng/نظرًا لاستقرارها وضمان عملها الرسمي.**
- **يستخدم baseline/إذا كنت بحاجة إلى ISO صغير جدًا وقابل للتخصيص بالكامل.**

3.نسخ الملف الشخصي releng

لبدء تخصيص توزيعتك الخاصة، قم بنسخ الملف الشخصي releng/إلى دليل عملك:

```
mkdir ~/my-archiso

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

cd ~/my-archiso/releng/
```

الآن يمكنك:

- **تعديل الحزم :حرر ملف packages.x86_64لإضافة أو إزالة الحزم.**
- **ضبط ملفات الإعداد :عدل الملفات الموجودة داخل airootfs/وprofiledef.sh.**
- **إضافة سكربتات أو أدوات مخصصة :دمج أي سكربتات أو أدوات خاصة بتوزيعتك.**

بعد إجراء التعديلات، يمكنك بناء ISO المخصص:

```
sudo mkarchiso -v .
```

خيار v-يفعّل الوضع التفصيلي (verbose) ليعرض الخطوات بالتفصيل أثناء عملية البناء.

هيكلية ملفاتArchiso

Archiso Configs

(/usr/share/archiso/configs/)

الملف/المجلد	الوصف
releng/	الإصدار المستقر الرسمي (نقطة البداية المثالية لتخصيصISO)
packages.x86_64	الحزم الأساسية للتوزيعة
airootfs/	نظام الملفات الافتراضي للISO
profiledef.sh	يحدد أذونات الملفات والإعدادات
...	ملفات إعداد أخرى
baseline/	نسخة صغيرة جدًا مع عدد قليل من الحزم، مناسبة للمشاريع الصغيرة
next/	النسخة التجريبية/التطويرية مع أدوات غير مستقرة

خطوات التخصيص العملية

1. **نسخ الملف الشخصي الرسمي:**

```
mkdir ~/my-archiso

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

cd ~/my-archiso/releng/
```

2. **تحرير الحزم:**

- **packages.x86_64: أضف أو أزل الحزم بما يتوافق مع هوية التوزيعة الخاصة بك.**

3. **تعديل نظام الملفات الجذري والإعدادات:**

- **airootfs/: أضف الملفات أو التطبيقات المخصصة هنا.**
- **profiledef.sh: اضبط الأذونات والإعدادات الخاصة بتوزيعتك.**

4. **بناء الـ ISO المخصص:**

```
sudo mkarchiso -v .
```

2.7.1تعديل ملف profiledef.sh

سنقوم بتقسيم هذا الملف إلى ثلاثة أقسام لتسهيل سير العمل وجعل العملية سلسة ومباشرة.

القسم 1: تعريفات ISO وطرق البناء

```
#!/usr/bin/env bash

# Suppress a specific warning from the ShellCheck tool (SC2034 indicates a defined but unused variable).

# shellcheck disable=SC2034
```

```
** #اسم الـISO:**

#يحدد الاسم النهائي لملف الـISO ، وهو الاسم الذي سيراه المستخدم عند التحميل أو التوزيع.

iso_name="Helwan-Linux-stable"
```

```
** #تسمية الـISO:**

#يتم كتابة هذه التسمية على الـ ISO عند الحرق أو عند الإقلاع من CD/USB.

iso_label="Helwan-Linux-stable-v1.1"
```

```
** #معلومات الناشر**:

#تظهر في بيانات الـ ISO الوصفية.(metadata)

iso_publisher="helwanlinux <helwanlinux@gmail.com>"
```

** #وصف التطبيق**:

#يصف الهدف من الـ .ISO في هذه الحالة، هو بيئة.Live/Rescue

iso_application="Helwan Linux Live/Rescue DVD"

** #نسخة الـ**:ISO

#رقم داخلي للإصدار للمطورين؛ عادة لا يظهر للمستخدم النهائي.

iso_version="v1.1"

** #دليل التثبيت**:

#يحدد اسم الدليل داخل الـ ISO الذي يحتوي على ملفات التثبيت الأساسية.

#في أنظمةArch ، يسمى عادة 'arch' لأن محملات الإقلاع (syslinux) أو (grub تبحث عنه.

install_dir="arch"

** #طرق البناء**:

#يحدد نوع المخرجات التي ستنتجها أداة.`mkarchiso`

#القيمة 'iso' تعني أن المخرجات ستكون ملف ISO قابل للإقلاع، وهو المعتاد للتوزيعات.

#يمكن إضافة أوضاع أخرى مثل 'netboot' أو 'bootstrap' لاستخدامات خاصة، لكنها أقل شيوعًا.

#مثال:

#إذا وضعت:

buildmodes=('iso' 'bootstrap')

#ستقوم أداة `mkarchiso` بإنتاج مخرجين:

1. #ملف ISO قابل للإقلاع.

2. #دليل bootstrap يحتوي على ملفات النظام الأساسية كأساس لبناء توزيعات أخرى.

buildmodes=('iso')

هذا القسم يحدد بيانات الـ ISO الأساسية ويبين كيف سيتم تنفيذ عملية البناء.

المعمارية المستهدفة

- arch="x86_64":يحدد المعمارية المستهدفة للـISO ، بحيث يكون متوافقًا مع الأنظمة 64-بت.

ملف إعداداتPacman

- pacman_conf="pacman.conf":يشير إلى ملف إعدادات Pacman المستخدم أثناء البناء، عادة يكون محليًا لتخصيص المستودعات وخيارات Pacmanالأخرى.

نوع صورةAIROOTFS

- airootfs_image_type="squashfs":يحدد نوع صورة نظام الملفات الجذري .(airootfs) استخدام squashfsيعني ضغط الملفات بصيغة SquashFSللقراءة فقط، مما يقلل الحجم ويزيد سرعة التحميل أثناء التشغيل.

خيارات أداة ضغطAIROOTFS

airootfs_image_tool_options=('-comp' 'xz' '-Xbcj' 'x86' '-b' '1M' '-Xdict-size' '1M')

- comp xz:-يستخدم خوارزمية ضغط.xz
- Xbcj x86:-يطبق فلترة Branch/Jump خاصة بمعمارية x86 لتحسين الضغط.
- b 1M:-حجم الكتلة 1 ميجابايت لتحسين الكفاءة.
- Xdict-size 1M:-حجم القاموس للضغط 1 ميجابايت.

ضغط أرشيفBootstrap

bootstrap_tarball_compression=('zstd' '-c' '-T0' '--auto-threads=logical' '--long' '-19')

- zstd:يستخدم خوارزمية Zstandard السريعة والفعالة.
- c:-يكتب البيانات المضغوطة إلى.stdout
- T0:-يستخدم جميع نوى المعالج المتاحة.
- auto-threads=logical:--يحدد عدد الخيوط تلقائيًا بناءً على عدد النوى المنطقية.
- long:--يُفعّل الوضع الطويل لأقصى كفاءة ضغط.
- 19:-أعلى مستوى ضغط متاح.

القسم 3: أذونات الملفات (file_permissions) في profiledef.sh

هذا القسم مهم جدًا لتحديد أذونات الملفات والمجلدات الحساسة أثناء بناء التوزيعة، لضمان عدم وصول المستخدمين العاديين إليها، مع السماح للنظام والمستخدمين بالوصول أو التنفيذ عند الحاجة أثناء الجلسات الحية أو التثبيت.

فهم أذونات الملفات

في أنظمةUnix ، تُمثل الأذونات برقم ثماني ثلاثي، حيث يمثل كل رقم الأذونات :المالك، المجموعة، والآخرين:

- **:4قراءة(r)**
- **:2كتابة(w)**
- **:1تنفيذ(x)**

مثال 750 :يعني:

- **المالك: 7 (4+2+1 = قراءة، كتابة، تنفيذ(**
- **المجموعة: 5 (4+1 = قراءة، تنفيذ(**
- **الآخرون: 0 (لا أذونات(**

في profiledef.sh، الصيغة عادة"owner:group:permissions" :، حيث 0:0تعني ملكيةroot.

تحديد الأذونات للملفات والمجلدات الأساسية

الوصف	الأذونات	الملف / المجلد
كلمات مرور المستخدمين مشفرة، للمالك فقط.	0:0:400	/etc/shadow
دليل المستخدمroot ، كامل للمالك، قراءة/تنفيذ للمجموعة، لا للآخرين.	0:0:750	/root
سكربت آلي للتثبيت، قابل للتنفيذ للجميع.	0:0:755	/root/.automated_script.sh
مفتاح التشفير الخاص، للمالك فقط.	0:0:700	/root/.gnupg
سكربت اختيار أفضل سيرفر، قابل للتنفيذ للجميع.	0:0:755	/usr/local/bin/choose-mirror
دليل تثبيت، قابل للتنفيذ للجميع، تعديل root فقط.	0:0:755	/usr/local/bin/Installation_guide
إدارة الصوت فيLive CD ، قابل للتنفيذ للجميع.	0:0:755	/usr/local/bin/livecd-sound
قواعد التحكم بالصلاحيات، للمالك والمجموعة فقط.	0:0:750	/etc/polkit-1/rules.d
إعدادات إضافية لـsudo ، للمالك والمجموعة فقط.	0:0:750	/etc/sudoers.d

أمثلة إضافية للتخصيص

- :etc/ssh/sshd_configأذونات 0:0:600، rootفقط للقراءة والكتابة.
- :etc/fstabأذونات 0:0:644، قابل للقراءة للجميع، كتابة للمالك فقط.
- :var/logأذونات 0:0:750، حماية سجلات النظام.
- :boot/grub/grub.cfgأذونات 0:0:600، حماية إعدادات.bootloader

بهذا نكون قد أنهينا استعراض ملف profiledef.shبالكامل، من تعريف بيانات الـISO ، تحديد المعمارية وخيارات الضغط، إلى ضبط أذونات الملفات الأساسية. هذه الإعدادات هي العمود الفقري لأي توزيعة مبنية علىArch ، لضمان أن يكون الـ ISO الناتج مستقرًا، آمنًا، وجاهزًا للاستخدام العملي.

في الأقسام القادمة، سننتقل من مرحلة التحضير إلى التنفيذ العملي خطوة بخطوة لتشكيل التوزيعة.

2.7.2إضافة الحزم في packages.x86_64

بدء العمل: تحديد نكهة التوزيعة والهوية التقنية

الخطوة الأساسية الآن هي تحديد نكهة التوزيعة الفريدة والهوية التقنية الخاصة بك. هذا هو ما يجذب المستخدمين المستهدفين. عند فتح ملف packages.x86_64، ستجد أن Arch Linux يملأه مسبقًا بالحزم الأساسية الصادرة شهريًا، لكنه لا يشمل مثبتات رسومية، أو بيئات سطح المكتب، أو حتى متصفحك المفضل. هنا يظهر دورك وخبرتك، لتعلن التوزيعة: "هذا هو الشيفرة الجينية الخاصة بي".

ملاحظة: إضافة رمز #في بداية السطر يجعل عملية بناء التوزيعة تتجاهل هذا السطر ومحتواه بالكامل.

```
#calamares-packages

calamares

hel-calamares-config

os-prober

ckbcomp

mkinitcpio-openswap
```

الحزم الأساسية المضافة

- :calamaresالمثبت الرسومي الشهير المستخدم في المئات من التوزيعات، مما يجعل عملية تثبيت التوزيعة سهلة للمستخدمين.
- :hel-calamares-configملفات التخصيص الخاصة بتوزيعةHelwan ، لتعديل مظهر وسلوك المثبت الرسومي بما يتماشى مع هوية التوزيعة.

- **os-prober: حزمة أساسية تساعد على اكتشاف أنظمة التشغيل الأخرى المثبتة مسبقًا، لتسهيل التثبيت الثنائي (Dual-Boot).**

تخصيصات إضافية

- **ckbcomp: أداة لتخصيص تخطيطات لوحة المفاتيح ودعم لغات معينة أو مفاتيح مخصصة.**
- **mkinitcpio-openswap: تضيف دعمًا لفتح أقسام swap المشفرة أثناء الإقلاع، مما يعزز أمان النظام.**

حزم الواجهة الرسومية (GUI Packages)

```
#GUI-packages##

xorg

xorg-apps

xf86-input-libinput

xdg-user-dirs

lightdm

lightdm-slick-greeter

base-devel

pacman-contrib

bash-completion

zenity
```

هذه الحزم تشمل الحد الأدنى لتشغيل واجهة رسومية كاملة على التوزيعة. بدون هذه المكونات، سيبقى النظام في وضع سطر الأوامر CLI فقط. الحزم التالية تشكل العمود الفقري للجزء الرسومي وتكمل الهوية التقنية للتوزيعة.

1. **xorg:**
   خادم X.Org الأساسي لتشغيل أي واجهة رسومية على Linux ، يدير العرض ويشغل النوافذ والتطبيقات الرسومية.
2. **xorg-apps:**
   مجموعة أدوات مساعدة لـ Xorg ، مثل:
   - **xrandr: لتغيير دقة الشاشة.**
   - **xset: لإعدادات الإدخال والإخراج.**
   - **xhost: للتحكم بالوصول إلى خادم X.**

3. **xf86-input-libinput:**
حزمة تعريفات الإدخال للفأرة ولوحات المفاتيح، تعتمد على مكتبة.libinput

4. **xdg-user-dirs:**
تنشئ مجلدات المستخدم القياسية مثلDocuments ، Downloads، .Pictures

5. **lightdm:**
مدير تسجيل الدخول الرسومي.(Display Manager)

6. **lightdm-slick-greeter:**
واجهة الترحيب لـLightDM ، تضيف شاشة تسجيل دخول حديثة وأنيقة.

7. **base-devel:**
مجموعة أدوات تطوير أساسية مثل gcc و make وautoconf ، ضرورية لتثبيت أو بناء البرامج من AUR أو من المصدر.

8. **pacman-contrib:**
أدوات إضافية لـpacman ، مثل:

- **checkupdates:للتحقق من التحديثات المتاحة.**
- **paccache:لإزالة الحزم القديمة من الكاش.**

9. **bash-completion:**
ميزة الإكمال التلقائي للأوامر في الطرفية، لتسهيل الكتابة وتسريعها.

10. **zenity:**
أداة لإنشاء نوافذ حوار رسومية من سكربتاتBash ، مثل عرض رسالة خطأ أو نافذة اختيار ملفات.

ملخص:
تعمل هذه الحزم معًا لتوفير الحد الأدنى لتشغيل واجهة رسومية احترافية على التوزيعة، من خادم العرض (xorg) إلى شاشة تسجيل الدخول(lightdm-slick-greeter) ، مع دعم الأدوات المساعدة للمستخدمين والمطورين.

حزم الشبكة وBluetooth

تركز هذه الحزم على الاتصال بالشبكة (سلكية/لاسلكية) ودعمBluetooth ، بالإضافة إلى تحسين إدارة الصوت والشبكة. بدونها، لن يتمكن المستخدمون من الاتصال بالإنترنت أو استخدام سماعات.Bluetooth

```
#Network & Bluetooth

networkmanager

network-manager-applet

bluez

bluez-utils
```

pipewire-pulse

reflector

1. **networkmanager: أداة إدارة الشبكات الأساسية، تدعم Wi-Fi ، Ethernet، VPN، وتبدأ تلقائيًا عند الإقلاع.**
2. **network-manager-applet: واجهة رسومية لـ NetworkManager ، تظهر في شريط النظام، لتسهيل اختيار الشبكات وإدارة VPNs.**
3. **bluez: البروتوكول الأساسي لدعم أجهزة Bluetooth.**
4. **bluez-utils: أدوات إضافية لإدارة Bluetooth عبر الطرفية، مثل bluetoothctl.**
5. **pipewire-pulse: طبقة توافق بين PipeWire وPulseAudio ، لدعم الصوت و Bluetooth بكفاءة أعلى.**
6. **reflector: لإدارة مرايا Arch Linux ، يمكن ترتيبها حسب السرعة أو آخر تحديث.**
   - **مثال عملي:**
7. **reflector --country "Egypt" --age 12 --sort rate --save /etc/pacman.d/mirrorlist**

ملخص:
تقدم هذه الحزم بنية أساسية للاتصال بالإنترنت وإدارة Bluetooth ، مع تحسين الصوت وضمان تحديث سريع للحزم.

حزم بيئة سطح المكتب

##desktop-environment

cinnamon

nemo-fileroller

evince

gedit

xfce4-terminal

gnome-calculator

eog

helfetch

gnome-keyring

توفر هذه الحزم بيئة سطح المكتب الأساسية وتطبيقاتها، من واجهة المستخدم إلى إدارة الملفات والوثائق والأدوات الصغيرة التي تجعل التوزيعة عملية وسهلة الاستخدام.

الحزم الأساسية:

1. **cinnamon: بيئة سطح مكتب حديثة، سهلة الاستخدام وقابلة للتخصيص.**
2. **nemo-fileroller: مدير الملفات مع دعم ضغط وفك الملفات مباشرة من قائمة النقر الأيمن.**
3. **evince: عارض مستندات خفيف الوزن لملفات PDF و.PostScript**
4. **gedit: محرر نصوص GNOME الافتراضي، بسيط ومستقر.**
5. **xfce4-terminal: محاكي طرفية خفيف وسريع، يدعم التبويبات والألوان.**
6. **gnome-calculator: آلة حاسبة بسيطة وقوية، تدعم الوضع العلمي والمبرمج.**
7. **eog (Eye of GNOME): عارض صور سريع وخفيف الوزن.**
8. **helfetch: أداة معلومات نظام مخصصة، مشابهة لـneofetch ، تعزز هوية التوزيعة.**
9. **gnome-keyring: مدير كلمات المرور والتوثيق، لتخزين كلمات Wi-Fi والمفاتيح بأمان.**

ملخص:
تحول هذه الحزم النظام من واجهة رسومية فارغة إلى بيئة سطح مكتب متكاملة، وتحدد الهوية المرئية والوظيفية للتوزيعة.

2.7.3 الأدوات والمستودعات(Utilities & Repositories)

بعد تغطية ملف pacman.conf في قسم سابق—حيث شرحنا كيفية تكوين المستودعات الرسمية—وعملنا مع الحزم الأساسية في bootstrap_packages.x86_64، من المفيد استعراض الأدوات والمستودعات الإضافية التي قد ترغب في تضمينها في التوزيعة الخاصة بك.

المستودعات(Repositories)

افتراضيًا، تعتمد Arch Linux على ثلاثة مستودعات رئيسية:

- **core: يحتوي على الحزم الأساسية للنظام.**
- **extra: يشمل تطبيقات سطح المكتب والأدوات الشائعة.**
- **community: يوفر برامج يتم صيانتها من قبل المجتمع.**
- **multilib) اختياري :(يدعم تشغيل التطبيقات 32-بت على أنظمة 64-بت.**

بالإضافة إلى ذلك، يمكنك إنشاء وإضافة مستودع مخصص خاص بك (مثل chaotic-aur) لاستضافة حزم التوزيعة الفريدة، العلامة التجارية، والأدوات الخاصة بك.

الأدوات(Utilities)

بعض الأدوات يمكن أن تحسن تجربة المستخدم وتوفر نظامًا أكثر تكاملاً، مثل:

- **git:للتحكم في النسخ واسترجاع الحزم من.AUR**
- **wgetأو :curl لتحميل الملفات من الإنترنت عبر الطرفية.**
- **nanoأو :vim محررات نصوص أساسية لتحرير الملفات.**
- **htop:لمراقبة العمليات والموارد بشكل تفاعلي.**
- **man-dbو :man-pages توفر توثيق الأوامر وصفحات المانيوال.**

ملخص:
إضافة هذه الأدوات والمستودعات تمنح توزيعتك أساسًا قويًا ومرنًا.
وبما أننا غطينا الأساسيات في الأقسام السابقة، سننتقل الآن إلى القسم 2.7.4 للتركيز على بيئة سطح المكتب والتطبيقات الرئيسية، حيث تبدأ الهوية البصرية والعملية للتوزيعة بالتشكل.

2.7.4استكشاف دليل usr/

بعد اختيار الحزم الأساسية، تكوين الأذونات، وإعداد بيئة التثبيت، حان الوقت للتعمق في العمود الفقري للتوزيعة :دليل .usr/
يعد هذا الدليل الموطن الأساسي لمعظم التطبيقات والمكتبات والأدوات التي تضمن تشغيل النظام بسلاسة بعد الإقلاع.

باختصار، يحتوي دليل usr/على كل ما يحتاجه المستخدم لتشغيل البرامج وبيئات سطح المكتب، مما يقلل الحاجة للوصول إلى دليل rootأو ملفات النظام الأساسية. الفهم والإدارة الدقيقة لهذا الدليل تمنحك القدرة على:

- **تخصيص التطبيقات :اختيار الحزم التي تعكس هوية التوزيعة وتلبي احتياجات المستخدمين.**
- **إدارة المكتبات :ضمان توافق المكتبات والأدوات مع الحزم الإضافية التي تخطط لتضمينها.**
- **تحسين تجربة المستخدم :زيادة كفاءة النظام سواء في بيئة سطح المكتب أو عبر سطر الأوامر.**

في الأقسام القادمة، سنحلل الدلائل الفرعية الأساسية داخل usr/، مع التركيز على ما يجب تضمينه وتعديله لإنشاء توزيعة قوية، آمنة، وسهلة الاستخدام.

2.7.4.1استكشاف دليل /usr/share/grub/themes/helwan

الآن بعد أن فهمنا هيكل وأهمية دليل usr/في التوزيعة، حان الوقت للغوص في مكون بصري مؤثر :ثيمات GRUB الخاصة بالتوزيعة.

يحتوي هذا الدليل على الملفات التي تحدد مظهر قائمة الإقلاع عند بدء تشغيل النظام. تصميم هذا الثيم ليس مجرد جمالية؛ بل يعكس هوية التوزيعة، مقدّمًا للمستخدم الانطباع الأول عن جودة النظام منذ اللحظة الأولى.

فهم محتويات هذا الدليل وكيفية تعديلها يمكّنك من:

- تخصيص واجهة القائمة الرسومية للإقلاع لتتماشى مع هوية Helwan Linux.
- التحكم بألوان، شعارات، وخطوط GRUB.
- تحسين تجربة المستخدم منذ اللحظة الأولى لتفاعله مع النظام.

في الأقسام القادمة، سنشرح الملفات الأساسية داخل هذا الدليل مع أمثلة عملية لتطبيق ثيم مخصص بالكامل.

شرح الأكواد الأساسية في theme.txt

يسمح هذا الملف بتخصيص واسع لمظهر محمل الإقلاع GRUB. فيما يلي أهم الأكواد ووظائفها:

1. الخيارات الرئيسية (Main Options)

- title-text: ""
  يحدد نص العنوان الذي يظهر في أعلى واجهة الإقلاع. تركه فارغًا يعني عدم عرض أي عنوان.
- desktop-image: "background.png"
  يحدد الصورة الخلفية لقائمة الإقلاع. يمكن استبدال "background.png" بأي صورة أخرى لتعكس هوية التوزيعة.
- desktop-color: "#000000"
  اللون المستخدم كخلفية إذا لم يتم عرض الصورة، هنا اللون أسود.
- terminal-font: "Terminus Regular 18"
  نوع وحجم الخط المستخدم في نافذة الطرفية ضمن GRUB.
- terminal-box: "terminal_box_*.png"
  شكل الإطار أو الصورة المحيطة بواجهة الطرفية.
- terminal-left, terminal-top
  تحدد موقع الإطار على الشاشة كنسبة مئوية من الحافة اليسرى والعلوية.
- terminal-width, terminal-height
  تحدد عرض وارتفاع نافذة الطرفية كنسبة مئوية من حجم الشاشة.
- terminal-border: "0"
  سمك الحدود حول نافذة الطرفية، 0 يعني بدون حدود.

2. قائمة الإقلاع (Boot Menu)

+ boot_menu { ... }

تحدد هذه القسم تصميم قائمة الإقلاع نفسها:

- left, top: موقع القائمة على الشاشة نسبة إلى العرض والارتفاع.
- width, height: حجم القائمة كنسبة مئوية من الشاشة.
- item_font: "Ubuntu Regular 20": الخط المستخدم لعناصر القائمة.
- item_color: "#cccccc": لون العناصر غير المحددة.
- selected_item_color: "#ffffff": لون العنصر المحدد حاليًا.
- icon_width, icon_height: أبعاد أي أيقونات مرتبطة بالعناصر.
- item_icon_space: المسافة بين الأيقونة والنص لكل عنصر.
- item_height, item_padding, item_spacing: ارتفاع العناصر، الحشوة الداخلية، والفواصل بين العناصر.
- selected_item_pixmap_style: "select_*.png": صورة أو نمط يُستخدم لتحديد العنصر المحدد بصريًا.

3. العد التنازلي (Countdown Label)

+ label { ... }

يحدد نص العد التنازلي الذي يظهر قبل بدء تشغيل النظام تلقائيًا:

- left, top: موقع النص على الشاشة.
- align: "center": توسيط النص.
- id: "timeout": معرف خاص للعد التنازلي.
- text: "Selected OS will start in %d seconds": نص العد التنازلي، حيث يتم استبدال %d بالثواني المتبقية.
- color: "#cccccc": لون النص.
- font: "Ubuntu Regular 17": نوع وحجم الخط للنص.

ملاحظة:
تتيح جميع هذه الإعدادات التحكم الكامل بمظهر قائمة GRUB ، من الخلفيات والخطوط إلى عرض العناصر المحددة والعد التنازلي. يمكن تعديل أي قيمة لتغيير شكل شاشة الإقلاع بما يتوافق مع هوية التوزيعة.

2.7.4.2 استكشاف دليل /usr/share/backgrounds

بعد تخصيص واجهة GRUB بصريًا، حان الوقت للغوص في عنصر أساسي لتجربة المستخدم الرسومية :خلفيات النظام.

يحتوي دليل /usr/share/backgrounds على الصور والخلفيات التي يمكن عرضها في بيئة سطح المكتب. تشمل هذه الخلفيات الافتراضية التي تظهر عند تسجيل الدخول أو الصور التي يمكن للمستخدمين اختيارها لاحقًا.

فهم محتويات هذا الدليل وكيفية تنظيمه يمكّنك من:

- **تخصيص الهوية البصرية للتوزيعة :اختيار الخلفيات التي تعكس شخصيةHelwan Linux ، مما يمنح المستخدمين شعورًا مميزًا منذ البداية.**
- **إدارة الأداء :استخدام صور بالحجم المناسب وبضغط ملائم يقلل من استهلاك الذاكرة ويسرع تحميل سطح المكتب.**
- **توفير خيارات للمستخدم :تمكين المستخدمين من تغيير الخلفيات بسهولة عبر إعدادات سطح المكتب.**

2.7.4.3استكشاف دليل /usr/local

بعد استكشاف الدلائل الأساسية مثل /usr/share، الذي يحتوي على ملفات النظام العامة، الخلفيات، وثيماتGRUB ، ننتقل الآن إلى /usr/local.
يعد هذا الدليل مهمًا لأي توزيعة مخصصة.

يخصص /usr/local لتثبيت البرامج والأدوات التي يضيفها المطور أو المستخدم نفسه، بعيدًا عن الحزم الرسمية التي يديرها مدير الحزم.(Pacman)

استخدام هذا الدليل يمكّنك من:

- **تخصيص التوزيعة بأدوات إضافية :تثبيت برامج أو سكريبتات محددة لتوزيعتك دون التأثير على النظام الأساسي للحزم.**
- **عزل البرمجيات المحلية عن النظام الأساسي :يقلل من خطر تعارض الحزم ويسهّل إدارة التحديثات.**
- **توفير بيئة مرنة للمستخدمين والمطورين :يمكن للمستخدمين إضافة أدواتهم، ويمكن للمطورين تجربة التطبيقات دون المخاطرة بمكتبات النظام الأساسية.**

في الأقسام القادمة، سنوضح كيفية تنظيم هذا الدليل، أهم الملفات التي يجب أن يحتويها، وكيفية إضافة أدوات توزيعتك بطريقة منظمة وآمنة.

2.7.4.3.1استكشاف دليل /usr/local/bin

يعد دليل /usr/local/binالقلب النابض للأدوات والبرامج الخاصة بالمستخدم أو المطور داخل التوزيعة.

بينما تحتوي الحزم الرسمية على برامج النظام في /usr/bin، فإن /usr/local/binمخصص للبرامج والسكريبتات التي تضيفها بنفسك أو تأتي مع التوزيعة بطريقة مخصصة، منفصلة عن مدير الحزم الرئيسي.(Pacman)

أهمية هذا الدليل:

- **تثبيت أدوات خاصة بالتوزيعة :كل سكريبت أو برنامج مصمم خصيصًا لتوزيعة Helwan Linux يمكن وضعه هنا لضمان سهولة الوصول والاستخدام.**
- **تسهيل الصيانة والتحديثات :فصل الأدوات المحلية عن برامج النظام يقلل التعارضات ويسهّل تحديث النظام أو البرامج.**
- **توفير بيئة تطوير مرنة :يسمح للمطورين والمستخدمين بتجربة سكريبتاتهم أو أدواتهم دون المخاطرة بمكتبات النظام الأساسية.**

في هذا القسم، سنشرح كيفية تنظيم هذا الدليل، إعداد الأذونات للأدوات والسكريبتات، وتقديم أمثلة عملية لأدوات التوزيعة التي يمكن وضعها هنا لتكون متاحة لجميع المستخدمين مباشرة.

استكشاف سكريبت /usr/local/bin/choose-mirror

هذا السكريبت، رغم بساطته، حيوي للتوزيعة.
يقوم بتحديث قائمة المرايا لمستودعات Arch Linux تلقائيًا أثناء الإقلاع، لضمان تنزيل الحزم بشكل أسرع وأكثر استقرارًا بناءً على موقع المستخدم.

1.هدف السكريبت

- **يحدد أي مرآة سيتم استخدامها لتحميل الحزم.**
- **يمكن تكوينه لاختيار المرايا تلقائيًا بناءً على الموقع أو معلمات سطر الأوامر.**
- **يحتفظ بنسخة أصلية من قائمة المرايا (mirrorlist.orig) لمنع فقدان البيانات.**

2.شرح الكود خطوة بخطوة

```
get_cmdline() {
local param
for param in $(</proc/cmdline); do
case "${param}" in
"${1}="*)
echo "${param##*=}"
return 0
;;
esac
done
return 1 # يشير للفشل إذا لم يتم العثور على المعامل
}
```

- **get_cmdline():دالة لقراءة سطر أوامر النواة عند الإقلاع من /proc/cmdline.**
- **تبحث عن قيمة معينة مرتبطة بمعامل محدد مثل mirror=.**
- **شرح تفصيلي:**

- for param in $(</proc/cmdline): يمر على كل كلمة موجودة في /proc/cmdline.
- case "${param}" in "${1}="*): يتحقق إذا بدأ المعامل الحالي بالمعامل المطلوب ($1.)
- echo "${param##*=}": إذا وجد، يستخرج ويعرض القيمة بعد علامة =.
- return 0: يخرج بنجاح بعد العثور على القيمة.

```
mirror="$(get_cmdline mirror)"
[[ "$mirror" == 'auto' ]] && mirror="$(get_cmdline archiso_http_srv)"
[[ -n "$mirror" ]] || exit 0
```

- السطر الأول: يحاول قراءة قيمة المرآة مباشرة من سطر الأوامر.
- السطر الثاني: إذا كانت القيمة auto، يحاول الحصول على عنوان السيرفر من archiso_http_srv.
- السطر الثالث: إذا لم يتم تحديد مرآة، يخرج السكريبت دون تعديل mirrorlist.

```
mv /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.orig
```

- إنشاء نسخة احتياطية من قائمة المرايا الحالية لتجنب فقدان الإعدادات الأصلية.

```
cat >/etc/pacman.d/mirrorlist <<EOF
#
# Arch Linux repository mirrorlist
# Generated by archiso
#

Server = ${mirror%%/}/\$repo/os/\$arch
EOF
```

- يكتب قائمة المرايا الجديدة في /etc/pacman.d/mirrorlist.
- ${mirror%%/}: إزالة أي شرطة مائلة في نهاية عنوان المرآة.
- \$repo\$ و \$arch\: متغيرات pacman القياسية لحل اسم المستودع والهندسة المعمارية تلقائيًا.

3. أهمية السكريبت

- تسريع تحميل الحزم ⚡: اختيار أقرب وأسرع مرآة يقلل وقت تنزيل الحزم.

- **المرونة :يمكن تحديد المرآة من سطر الأوامر أثناء الإقلاع.**
- **الأمان :□الاحتفاظ بنسخة أصلية يحمي النظام من فقدان الإعدادات.**
- **تجربة المستخدم :يسهل عملية الإعداد الأولية، خاصة في بيئات.Live CD/USB**

4.أمثلة عملية

- **لتحديد مرآة أثناء الإقلاع:**

mirror=http://mirror.example.com

- **للسماح للنظام باختيار السيرفر تلقائيًا:**

mirror=auto

استكشاف سكريبت launch

مسؤول عن تشغيل المثبت الرسومي Calamares بشكل صحيح في بيئة التوزيعة المباشرة(Live) ، خصوصًا عند استخدام ميزة Copy-to-RAM.

1.تعريف المتغيرات الأساسية

DIR="/etc/calamares"

KERNEL=`uname -r`

- **DIR:مسار ملفات تكوين.Calamares**
- **KERNEL:نسخة النواة الحالية، ضرورية لتحديد المسار الصحيح للنواة أثناء التثبيت.**

2.التحقق منCopy-to-RAM

if [[-d "/run/archiso/copytoram"]]; then

- **يتحقق إذا تم الإقلاع باستخدام .copytoram إذا كان موجودًا، يجب تعديل المسارات داخل.Calamares**

3.تعديل مسارات الملفات في unpackfs.conf

sudo sed -i -e 's|/run/archiso/bootmnt/arch/x86_64/airootfs.sfs|/run/archiso/copytoram/airootfs.sfs|g' "$DIR"/modules/unpackfs.conf

sudo sed -i -e "s|/run/archiso/bootmnt/arch/boot/x86_64/vmlinuz-linux|/usr/lib/modules/$KERNEL/vmlinuz|g" "$DIR"/modules/unpackfs.conf

- **تعديل مسار airootfs.sfsالموجود في الذاكرة.**
- **تحديث مسار النواة لضمان استخدام Calamares للنواة الصحيحة.**

4.تشغيلCalamares

sudo -E calamares -d

- **:sudo -Eيحتفظ بالمتغيرات البيئية عند التشغيل كروت.**
- **:d-تشغيل Calamares في وضع التصحيح.**

5.أهمية السكريبت

- **دعمCopy-to-RAM.**
- **منع أخطاء المسارات.**
- **تسهيل الصيانة والتطوير.**
- **تجربة مستخدم سلسة عند الإقلاع من USB/CD مباشر.**

استكشاف سكريبت /usr/local/bin/livecdsound

سكريبت مهم لإدارة الصوت في بيئة Live CD/USB لتوزيعةHelwan Linux.

1.الهدف الرئيسي

- **فك كتم الصوت لجميع بطاقات الصوت المتصلة.**
- **ضبط مستويات الصوت للقنوات الأساسية(Front, Master, Headphone, PCM, DAC, Synth).**
- **اختيار بطاقة الصوت الرئيسية إذا كانت متعددة.**
- **توفير إشارات صوتية لتسهيل اختيار الجهاز الافتراضي.**
- **ضمان التوافق مع أنواع مختلفة من البطاقات(Intel HDA, SB Live, VIA).**

2.المكونات والوظائف الأساسية

أ. الدوال المساعدة

- **:()usageعرض كيفية استخدام السكريبت وخياراته.**
- **:()bugoutطباعة رسالة خطأ عند استدعاء غير صحيح.**
- **:()echo_card_indicesقراءة أرقام جميع بطاقات الصوت من /proc/asound/cards.**

ب. التحكم بالصوت لكل بطاقة

- **:(card, control, level)unmute_and_set_levelفك كتم قناة معينة وضبط مستوى الصوت.**
- **:(card, control)mute_and_zero_levelكتم قناة وضبط مستوى الصوت إلى 0%.**

- **switch_control(card, control, state): تشغيل أو إيقاف التحكم (مثل.Master Playback Switch)**
- **sanify_levels_on_card(card): ضبط مستويات افتراضية لضمان عمل الصوت بشكل مستمر.**
- **sanify_levels(card|all): تطبيق sanify_levels_on_card على بطاقة واحدة أو جميع البطاقات.**
- **list_non_pcsp_cards(): استرجاع قائمة البطاقات القادرة على PCM مع استبعاد مكبرات الصوت الداخلية.**
- **unmute_all_cards(): فك كتم جميع البطاقات.**

ج. إدارة البطاقة الافتراضية

- **is_numeric(): التحقق إذا كان الإدخال رقم بطاقة.**
- **set_default_card(card): تحديث /etc/asound.conf بالبطاقة الافتراضية.**
- **play_on_card(card, file): تشغيل ملف صوتي على بطاقة معينة.**
- **pick_a_card(): اختيار البطاقة الافتراضية بين عدة بطاقات مع إشارات صوتية.**

.3خيارات وقت التشغيل

الخيار	الوظيفة
-h / --help	عرض رسالة المساعدة
-u / --unmute	فك كتم جميع البطاقات فورًا
-p / --pick	اختيار البطاقة الرئيسية للتشغيل

- **إذا لم يتم تمرير خيار أو تم تمرير خيار غير مدعوم، يطبع السكريبت خطأ ويخرج.**

.4أمثلة عملية

1. **فك كتم جميع البطاقات:**

```
livecdsound -u
```

2. **اختيار بطاقة للتشغيل:**

```
livecdsound -p
```

3. **عرض رسالة المساعدة:**

```
livecdsound -h
```

.5الأهمية في التوزيعة

- **صوت فوري بعد الإقلاع من Live CD/USB.**

- **مرونة للمستخدم باختيار البطاقة وضبط الصوت.**
- **دعم بطاقات مختلفة.(Intel HDA, SB Live, VIA)**
- **سهولة الصيانة عبر وضع السكريبت في .usr/local/bin/**

ملاحظة :يمكن دمج السكريبت مع تسلسل بدء التشغيل لتوفير صوت جاهز من لحظة الإقلاع، مما يحسن تجربة المستخدم منذ البداية.

الخلاصة: المحتويات الرئيسية لدليل usr/

بعد استكشاف الدلائل الفرعية الرئيسية داخل usr/، بما في ذلك:

- **/usr/share/grub/themes/helwan**
- **/usr/share/backgrounds**
- **/usr/local/bin**

يتضح أن هذا الدليل يشكل العمود الفقري لبيئة المستخدم في التوزيعة.

النقاط الرئيسية:

- **التخصيص والهوية :الملفات في usr/share/مثل ثيمات GRUB والخلفيات تحدد الهوية البصرية لـ Helwan Linux وتشكل الانطباع الأول للمستخدمين.**
- **أدوات المستخدم والمطورين usr/local/bin/ :يحتوي على سكريبتات وأدوات مثل choose-mirror, launch, و livecdsoundلتعزيز الوظائف وتوفير المرونة والحفاظ على فصلها عن ملفات النظام الأساسية.**
- **كفاءة النظام :تنظيم usr/بشكل صحيح يضمن عمل بيئة سطح المكتب، التطبيقات، وأدوات النظام بسلاسة دون لمس ملفات rootالحرجة.**
- **قابلية التوسع :فهم وإدارة usr/يمنحك القدرة على إضافة أو تعديل أو إزالة المكونات لتخصيص النظام حسب احتياجات المستخدمين.**

باختصار، إتقان محتويات وهيكلية usr/أمر حاسم لبناء توزيعة متينة، متناسقة بصريًا، وسهلة الاستخدام، حيث يعتبر المركز الرئيسي للتطبيقات والمكتبات والأدوات الأساسية.

2.8الملفات الأساسية في:etc/

2.8.1 linux.preset

عند بناء توزيعة لينكس، خصوصًا لبيئةLive ، يعتبر) initramfsنظام الملفات المؤقت الأولي في الذاكرة) عنصرًا حيويًا. هذا الملف مسؤول عن تحميل النواة (kernel) وتجهيز النظام المبدئي قبل الإقلاع الكامل. برنامج mkinitcpioيدير إنشاء هذه الصور، وسلوكه يتم تكوينه عبر ملفات.preset

أحد أهم هذه الملفات هو .etc/mkinitcpio.d/linux.preset/هذا الملف يُنسخ كما هو إلى صورة الـ ISO النهائية.

هدف ملفlinux.preset

يؤدي ملف linux.preset عدة وظائف رئيسية:

- تحديد النواة المستهدفة :يحدد أي نواة سيتم إنشاء initramfs لها.
- ربط النواة بالتكوين :يربط النواة بملف mkinitcpio.confالمقابل لها، والذي يحتوي على قائمة الوحدات (modules) والـ hooks المطلوبة للـ.initramfs
- تحديد أسماء ومسارات الملفات الناتجة :يوضح اسم ومكان ملف initramfs الذي سيتم إنشاؤه ووضعه في دليل .boot/
- توفير preset جاهز للاستخدام :يقدم إعدادًا مسبقًا يمكن استدعاؤه بأمر مثل .mkinitcpio -p archiso

مثال على التكوين

قد يبدو ملف linux.preset النموذجي كما يلي:

```
PRESETS=('archiso')

ALL_kver='/boot/vmlinuz-linux'

ALL_config='/etc/mkinitcpio.conf'

archiso_image="/boot/initramfs-linux.img"
```

- :ALL_kverيشير إلى النواة المستهدفة (في هذا المثال .(linux-/boot/vmlinuz
- :ALL_configيحدد ملف التكوين mkinitcpio الذي سيتم استخدامه.
- :archiso_imageيحدد اسم ملف initramfs الناتج.

التعامل مع تغييرات النواة

في بعض التوزيعات، قد لا تستخدم النواة الافتراضية linux فقط، بل قد تختار بدائل مثل:

-) linux-ltsدعم طويل المدى(
-) linux-zenمحسنة للأداء(

- linux-hardened (محسنة للأمان

إذا انتقلت إلى نواة مختلفة، يجب تعديل متغير ALL_kver ليشير إلى مسار النواة الجديد. مثال:

```
ALL_kver='/boot/vmlinuz-linux-lts'

lts_image="/boot/initramfs-linux-lts.img"
```

عواقب سوء التكوين

- عدم توافق النواة :إذا غيرت النواة ونسيت تحديث ملف linux.preset ، من المرجح أن يفشل النظام في الإقلاع، لأن initramfs الناتج لن يكون متوافقًا مع النواة الجديدة.
- البرامج المفقودة :تأكد من تضمين حزم النواة اللازمة (مثل linux-lts و linux-lts-headers) داخل بيئة بناء الـ ISO ، وإلا فلن يتمكن mkinitcpio من إنشاء initramfs.

اختيار النواة وتوافق الأجهزة

تغيير النواة يمكن أن يؤثر مباشرة على توافق الأجهزة:

- linux-lts: يوفر استقرارًا أكبر ودعم طويل المدى، مناسب للأجهزة القديمة.
- linux-zen: يهدف لأداء أعلى، لكنه قد يأتي أحيانًا على حساب الاستقرار.
- linux-hardened: يوفر أمانًا محسّنًا، لكنه قد يفرض قيودًا على بعض وظائف الأجهزة.

الخلاصة

ملف linux.preset هو الرابط الأساسي بين النواة و initramfs. أي تعديل على نوع النواة يجب أن ينعكس في هذا الملف.

- نسيانه :يؤدي إلى ISO غير قابل للإقلاع.
- تكوينه بشكل صحيح :ينتج ISO مخصص ومستقر لتوزيعتك.

2.8.2 /airootfs/etc/skel

مجلد (skel اختصار لـ – skeleton الهيكل العظمي) يعمل كقالب لبيئات المستخدمين الجدد. بمعنى آخر، يوفّر البنية الأساسية التي يتم بناء كل حساب مستخدم جديد عليها.
أي ملف أو مجلد يوضع داخل skel يتم نسخه تلقائيًا إلى مجلد المنزل للمستخدم الجديد (/home/username) عند إنشاء هذا المستخدم على النظام.

تتيح هذه الخاصية للمطورين والموزعين إمكانية تخصيص تجربة المستخدم من البداية من خلال:

- إدراج ملفات إعدادات أساسية (dotfiles) مثل bashrc. أو.profile.
- توفير مستندات ترحيبية أو تعليمية.
- ضبط إعدادات سطح المكتب أو البيئة الافتراضية.

2.8.3 .bashrc

مقدمة

ملف bashrc.هو واحد من أهم ملفات الإعدادات الخاصة بقشرة .Bash عند فتح المستخدم لجلسة شل تفاعلية جديدة، يتم تنفيذ محتوى هذا الملف تلقائيًا.
الهدف الأساسي منه هو تخصيص بيئة الطرفية بما يتوافق مع احتياجات المستخدم أو هوية التوزيعة.

الغرض من الملف

يُستخدم bashrc. لعدة وظائف رئيسية:

- **إعداد البيئة :ضبط متغيرات البيئة بما يتناسب مع الاحتياجات الخاصة.**
- **تخصيص المظهر :تعديل شكل ومظهر مؤشر الأوامر.(prompt)**
- **تحسين الاستخدام :إضافة) aliases اختصارات للأوامر) ووظائف لتسهيل سير العمل.**
- **تحسين الإخراج :تفعيل ميزات مثل ألوان الطرفية أو تعديل طريقة عرض مخرجات الأوامر.**

مثال عملي (من(Helwan Linux

#إذا لم تكن الجلسة تفاعلية، لا تفعل شيئًا

```
[[ $- != *i* ]] && return
```

#تفعيل ألوان الأمرls

```
alias ls='ls --color=auto'
```

#تخصيص الـprompt

```
PS1='[\u@\h \W]\$ '
```

#عرض إصدار التوزيعة منetc/os-release/

```
alias version="sed -n 1p /etc/os-release && sed -n 12p /etc/os-release && sed -n 13p /etc/os-release"
```

#اختصارات أوامرpacman

```
alias sync="sudo pacman -Syyy"
```

```
alias install="sudo pacman -S"

alias update="sudo pacman -Syyu"

alias search="sudo pacman -Ss"

alias search-local="sudo pacman -Qs"

alias pkg-info="sudo pacman -Qi"

alias local-install="sudo pacman -U"

alias clr-cache="sudo pacman -Scc"

alias unlock="sudo rm /var/lib/pacman/db.lck"

alias remove="sudo pacman -R"

alias autoremove="sudo pacman -Rns"
```

#اختصار معلومات النظام

```
alias helwan="uname -a"
```

أهمية الملف في التوزيعة

- **تأسيس هوية التوزيعة :إدراج أوامر فريدة مثل helwan أو version يعكس هوية التوزيعة.**
- **تحسين تجربة المستخدم :يوفّر أوامر جاهزة للاستخدام للمستخدمين الجدد لتسهيل التفاعل مع النظام.**
- **تقليل الأخطاء البشرية :يقلل الأخطاء مثل نسيان الخيارات الضرورية عند استخدام مدراء الحزم مثل.pacman**

اقتراحات للتخصيص

يمكن للمطورين توسيع ملف bashrc. لتقديم تجربة أغنى، مثل:

- **تخصيص متقدم للموجه:(Prompt)**
  - **عرض الوقت والتاريخ الحالي.**
  - **إظهار حالة فرع Git عند العمل على مشاريع.**
- **إضافة وظائف مخصصة:**
  - **إنشاء وظيفة مثل update-all التي تجمع بين تحديث النظام وتنظيف الكاش.**
- **تحسين نظام الألوان:**
  - **تلوين مخرجات أوامر مثل grep و diff و man لتحسين القراءة.**

- **التكامل مع أدوات أخرى:**
  - **تمكين fzf للبحث التفاعلي.**
  - **استبدال ls بـ exa لقائمة ملفات أكثر حداثة.**
- **إعدادات الأمان:**
  - **إعداد alias يحذر المستخدم قبل تنفيذ.rf- rm**

الخلاصة

ملف bashrc. ليس مجرد ملف إعدادات، بل هو أداة أساسية لتشكيل هوية التوزيعة وتحسين تجربة المستخدم. من خلال الاستفادة منbashrc. ، يمكن جعل الطرفية أكثر قوة ومرونة وأمانًا، مما يزيد من قيمة التوزيعة للمستخدم النهائي.

2.8.4 airootfs/etc/skel/.cinnamon/configs/menu@cinnamon.org

تخصيص التوزيعة لا يقتصر على ملفات الطرفية مثل bashrc. أوprofile. ؛ بل يمتد أيضًا إلى بيئة سطح المكتب.(DE)
نظرًا لأن Cinnamonهي بيئة سطح مكتب شائعة توفر تجربة رسومية قريبة من تجربة المستخدم العادية، فإن ملفات إعدادها تلعب دورًا أساسيًا في هوية التوزيعة.

ملفات رئيسية

menu@cinnamon.org

- **مسؤول عن إعداد قائمة التطبيقات في.Cinnamon**
- **يحتوي على تفاصيل مثل:**
  - **شكل أيقونة القائمة في شريط المهام.**
  - **النصوص أو العناوين الافتراضية.**
  - **بعض خيارات تخصيص المظهر.**

إدراج هذا النوع من التخصيص داخل مجلد skel يضمن أن كل مستخدم جديد يحصل على نفس المظهر والتجربة التي تعكس هوية التوزيعة دون الحاجة لإعادة ضبط يدوي.

ملف 0.jsonهو ملف إعدادات بصيغة JSON لتطبيق applet الخاص بالقائمة (menu@cinnamon.org) في.Cinnamon موقعه:

\airootfs\etc\skel\.cinnamon\configs\menu@cinnamon.org

يعني أن أي مستخدم جديد يُنشأ في التوزيعة سيرث هذه الإعدادات تلقائيًا، وبالتالي ستكون واجهة قائمة Cinnamon موحدة ومتوافقة مع هوية التوزيعة.

الشرح خطوة بخطوة

.1الهيكل العام
الملف منظم على شكل: صفحات ← أقسام ← مفاتيح.

- **layout: يحدد الصفحات الرئيسية.(Panel, Menu)**
- **كل صفحة تحتوي على أقسام (مثل المظهر والسلوك.(**
- **كل قسم يحتوي على مفاتيح تمثل الإعدادات الفعلية.**

.2قسم) Panel شريط المهام(
) panel-appearالمظهر:(

- **menu-custom: تمكين أو تعطيل استخدام أيقونة وعلامة مخصصة.**
- **menu-icon: تحديد أيقونة القائمة (يمكن استبدالها بشعار.(Helwan**
- **menu-icon-size: حجم الأيقونة (افتراضي: 32.(px**
- **menu-label: النص بجانب الأيقونة (محدد حاليًا بـ.(Helwan**

النقطة الأساسية: من أول إقلاع، يرى المستخدمون تسمية Helwan وأيقونة مخصصة، مما يعزز الهوية البصرية للتوزيعة.

) panel-behaveالسلوك:(

- **overlay-key: مفتاح الاختصار لفتح القائمة (عادة.(Super**
- **activate-on-hover: فتح القائمة عند مرور الماوس.**
- **hover-delay: التأخير قبل الفتح عند المرور.**
- **force-show-panel: يضمن ظهور الشريط حتى إذا كان مخفيًا تلقائيًا.**
- **enable-animation: يضيف تأثيرات فتح/إغلاق (معطلة افتراضيًا.(**

.3قسم) Menu القائمة نفسها(
) menu-layoutالتخطيط والمحتوى:(

- **show-category-icons: عرض أيقونات الفئات.**
- **show-application-icons: عرض أيقونات التطبيقات.**
- **category-icon-size & application-icon-size: التحكم في أحجام الأيقونات.**
- **favbox-show: عرض المفضلات وأزرار الجلسة.**
- **fav-icon-size & favbox-min-height: تحديد مظهر وأبعاد المفضلات.**
- **show-places: عرض العلامات المرجعية ,Documents, Downloads) إلخ.(**
- **show-recents: عرض الملفات المستخدمة مؤخرًا.**

- **menu-editor-button: يفتح أداة محرر القائمة.**

menu-behave (السلوك:)

- **enable-autoscroll: التمرير السلس في قائمة التطبيقات.**
- **search-filesystem: تمكين البحث في مسار نظام الملفات (معطل افتراضيًا.)**

القيمة المضافة للتوزيعة .4
:يضمن أن skel وضع هذا الملف داخل مجلد
.المخصصة Helwan كل مستخدم جديد يبدأ بنفس واجهة •
.موجود من أول تسجيل دخول (”Helwan“ الشعار (الأيقونة + العلامة •
.المفضلات وتصميم القائمة يظلان متسقين بين المستخدمين ← استقرار بصري •

.ليس مجرد بيئة افتراضية، بل هو أيضًا انعكاس لهوية التوزيعة Cinnamon وبالتالي، فإن سطح المكتب

:(Helwan Linux اقتراحات إضافية (لتعزيز العلامة التجارية لـ
.(/usr/share/icons مخصصة (من Helwan بأيقونة Cinnamon استبدال الأيقونة الرمزية الافتراضية لـ •
.لتحسين الرؤية (pxأو 40 pxزيادة حجم الأيقونة قليلاً (مثل 36 •
.”Helwan Start“ أو ”Helwan Menu“ إلى شيء أوضح مثل ”Helwan“ تغيير العلامة من •
.تمكين الرسوم المتحركة لتوفير تجربة مستخدم أكثر سلاسة •
.إذا كانت الخصوصية أولوية ”Recents“ تعطيل •

الخلاصة
كيف يمكن لمطوري التوزيعة تشكيل الانطباع الأول للمستخدم menu@cinnamon.orgيوضح هذا المثال لتخصيص •
.وتجربته العامة من خلال ضبط بيئة سطح المكتب بدقة
، ينطبق نفس المبدأ على الجوانب الأخرى من سطح المكتب: Cinnamon بينما يركز هذا الإعداد على قائمة تطبيقات •
.السمات، الأيقونات، الإضافات، وحتى سلوك مساحة العمل
وتقديم تجربة Helwan Linux باختصار، هذا مجرد نقطة انطلاق. يمكن إضافة تخصيصات أكثر تقدماً ودقة لتعزيز هوية •
.مستخدم احترافية جاهزة من الصندوق

(ISO نشر التوزيعة (بناء ونشر ملف 2.9

قابل للإقلاع يمكن ISO ، الخطوة الطبيعية التالية هي إنتاج ملفarchiso بعد الانتهاء من تخصيص التوزيعة باستخدام .اختباره، توزيعه، أو نشره للمستخدمين. هذه المرحلة تمثل الانتقال من مشروع تطوير محلي إلى منتج قابل للمشاركة

1. إنشاء ملف ISO باستخدام archiso

باستخدام ISO ، يمكنك إنشاء ملفreleng/archiso/~ في المسار helwanlinux افترض أنك أعددت مجلد المشروع باسم :الأمر التالي

```
cd ~/archiso/helwanlinux
sudo mkarchiso -v -o ~/iso-out .
```

• -v: يُمكّن من عرض التفاصيل الكاملة لعملية البناء.
• -o ~/iso-out: يحدد مجلد الإخراج لملف ISO.
• . يشير إلى أن البناء سيتم من الدليل الحالي.

بعد الانتهاء، ستجد ملف ISO مشابه لهذا:

```
~/iso-out/helwanlinux-2025.09.14-x86_64.iso
```

---

2. اختبار ISO محليًا

من الضروري اختبار التوزيعة قبل نشرها، ويمكن القيام بذلك باستخدام آلة افتراضية:

باستخدام QEMU/KVM:

```
qemu-system-x86_64 -boot d -cdrom ~/iso-out/helwanlinux-2025.09.14-x86_64.iso -m 2048
```

أو باستخدام VirtualBox:
• إنشاء آلة افتراضية جديدة.
• اختيار ملف ISO كمصدر للإقلاع.

3. توزيع ISO

بعد الاختبار، يمكنك رفع ملف ISO على منصات مختلفة:
• GitHub Releases: مثالي للمشاريع مفتوحة المصدر مع مستودع GitHub.
• SourceForge: خيار كلاسيكي لتوزيعات Linux.
• Google Drive / Mega / Nextcloud: للمشاركة السريعة مع مجموعة صغيرة.

⚠ دائمًا ارفع ملف SHA256 (مثل sha256sum) مع تحقق من صحة الملف لضمان سلامة الملف بعد التنزيل.

إنشاء ملف SHA256:

```
sha256sum helwanlinux-2025.09.14-x86_64.iso > SHA256SUMS
```

4. توثيق ما قبل النشر

:النجاح في النشر لا يقتصر على الملف التنفيذي فقط؛ بل يعتمد أيضًا على توثيق جيد
.مختصر: يوضح ميزات التوزيعة README •
.(آخر DE أو أي Cinnamon) لقطات شاشة: تعرض بيئة سطح المكتب •
.تعليمات التثبيت والإقلاع •
.سجل التغييرات: يوثق الجديد في كل إصدار •

---

5. التفاعل مع المنتديات والمجتمع

.المجتمع عنصر أساسي لأي مشروع مفتوح المصدر
:شارك مشروعك على
.Arch Linux منتديات •
• Reddit (r/linux, r/archlinux).
.(لإدراج التوزيعة لاحقًا) Distrowatch •

:نصائح للتفاعل
.(توقع النقد (أحيانًا قد يكون حادًا •
.خذ الملاحظات التقنية على محمل الجد وحاول التحسين •
.تجاهل التعليقات غير المفيدة •
."اعتمد مبدأ: "أصدر مبكرًا، وأصدر كثيرًا •

الخلاصة

قابل للإقلاع والنشر. من هنا، تبدأ الرحلة ISO تمثل هذه الخطوة نهاية المرحلة الأولى: بناء وتخصيص التوزيعة وإنتاج ملف .، الذي سنتناوله في الفصل التاليCalamares الأوسع: جعل التوزيعة سهلة التثبيت للمستخدم النهائي. وهنا يأتي دور مثبت

تحذير مهم للنشر ⚠
:عند مشاركة مشروعك في المنتديات العامة، تذكر أن طبيعة التفاعل تختلف من مجتمع لآخر

• الرسمي: معروف بالصرامة والدقة. يجب أن تكون الأسئلة موثقة ومكتوبة بوضوح؛ وإلا قد تواجه Arch Linux منتدى ردودًا حادة.
• مجتمع أوسع وأكثر تنوعًا، مناسب للنقاشات العامة وجمع الآراء، لكنه يتطلب :(r/archlinux و r/linux مثل) Reddit قبول النقد.
• LinuxQuestions (LQ): منتدى قديم ومنتشر. قد تجد بعض الردود قديمة، غير مفيدة، أو حتى محبطة عند عرض توزيعة جديدة. من الأفضل أن تكون مستعدًا لذلك وألا تعتمد عليه كمصدر رئيسي للتغذية الراجعة البناءة.

نصيحة: أو صفحة المشروع) أو المنصات GitHub Issues ركز دائمًا على المجتمعات التقنية المرتبطة بتوزيعتك (مثل بهذه الطريقة، ستحصل على ملاحظات عملية من مستخدمين ومطورين ذوي .(Matrix، Mastodon، Reddit) النشطة خبرة ذات صلة بمشروعك.

3.1مقدمةCalamares

1.أهمية تجربة التثبيت في أنظمة التشغيل

منذ فجر أنظمة التشغيل الحديثة، كان المثبّت (Installer) هو البوابة الأولى للمستخدم للدخول إلى عالم النظام. الانطباع الأول دائمًا حاسم: إذا كانت عملية التثبيت معقدة أو غامضة، فقد تثبط عزيمة المستخدم عن الاستمرار، حتى لو كان النظام نفسه غنياً بالاحتمالات. في المقابل، يزرع المثبت الجيد شعوراً بالثقة والأمان، مما يمهد الطريق لعلاقة طويلة الأمد بين المستخدم والتوزيعة.

في عالم لينكس تحديداً، حيث تتنوع التوزيعات بشكل لا يُصدّق، تصبح تجربة التثبيت معياراً أساسياً للتمييز. فبعض التوزيعات تستهدف مستخدمي الخوادم أو الهواة المتمرسين وتكتفي بمثبت نصي بسيط. بينما تسعى توزيعات أخرى، تهدف إلى جمهور أوسع، لتوفير مثبت رسومي كامل (Graphical Installer) يسهّل الدخول على المبتدئين.

2.المناهج التاريخية: من المثبتات النصية إلى الرسومية

- مُثبّت دبيان :(Debian Installer) مثبت نصي قديم ولكنه شديد المرونة، يسمح بتخصيص شبه كامل لكل مكون من مكونات النظام. على الرغم من قوته، لا تزال تجربته تعتبر معقدة لغير الخبراء.
- أناكوندا) (Anaconda) مُستخدم في Red Hat و :(Fedora يركز هذا المثبت على الثراء الوظيفي ويدعم السيناريوهات المعقدة مثل التثبيت عبر الشبكة، والتخزين المتقدم، والتشفير. ومع ذلك، قد يبدو ضخماً أو بطيئاً في بعض الحالات.
- ياست) (YaST) في :(openSUSE يوفر تجربة متكاملة لا تشمل التثبيت فحسب، بل إدارة النظام بعد التثبيت أيضاً، ويُعد مثالاً على عمق التصميم الألماني.
- يوبيكويتي) (Ubiquity) في :(Ubuntu ربما يكون الأكثر شهرة بين المستخدمين الجدد، حيث قدم منذ عام 2006 تجربة بصرية مبسطة جداً، مع عرض شرائح (slideshow) أثناء التثبيت لتعريف المستخدم بالنظام.

وسط هذا التنوع، ظهرت الحاجة إلى مثبّت مرن، خفيف الوزن، وقابل لإعادة الاستخدام عبر توزيعات متعددة. هنا، ظهر Calamares لسد هذه الفجوة.

3.ما هو Calamares ولماذا هو مختلف؟

Calamaresليس مجرد برنامج قائم بذاته؛ إنه إطار عمل (Framework) لتثبيت الأنظمة. هيكله يعتمد على الوحدات(Modules) ، مما يسمح لكل توزيعة باختيار وترتيب الوحدات التي تناسب احتياجاتها. هذا التصميم النمطي (Modular) يتيح للتوزيعة أن:

- تستخدم التثبيت التلقائي أو اليدوي.
- تحدد طرق تقسيم الأقراص (يدوي، تلقائي، LUKS، LVM، .(Btrfs
- تضيف المستخدمين وتضبط كلمات المرور.
- تثبت الحزم الأساسية أو الإضافية.
- تشغل سكربتات (scripts) ما بعد التثبيت.

الميزة العظمى لـ Calamares تكمن في كونه غير مرتبط بتوزيعة معينة. فهو يعمل مع آرتش لينكس(Arch Linux) ، و دبيان (Debian)، و فيدورا(Fedora) ، و openSUSE، وحتى التوزيعات المستقلة. وبالتالي، فقد أصبح بمثابة معيار مفتوح لتوفير تجربة تثبيت موحدة ولكن قابلة للتخصيص.

4.العلاقة بين Calamares وهوية التوزيعة

اختيار Calamares لا يعني التخلي عن الطابع الفريد لكل توزيعة. على العكس، يمكن للتوزيعة أن تعكس هويتها من خلال:

- **العلامة التجارية البصرية :(Visual Branding) الشعارات، الألوان، الخلفيات، ورسائل الترحيب.**
- **خيارات التثبيت الافتراضية :مثل التشفير الافتراضي، أنظمة الملفات(ext4 vs. Btrfs) ، أو تضمين حزم معينة.**
- **سيناريوهات التثبيت المتعددة :على سبيل المثال، "تثبيت مكتبي" مقابل "تثبيت حد أدنى."**

بالنسبة لحلوان لينكس(Helwan Linux) ، يمثل Calamares أداة استراتيجية: فهو يظهر للمستخدم أن التوزيعة ليست مجرد إعادة تجميع لحزم آرتش، بل هي مشروع متكامل له رؤية فريدة، وواجهة، وتجربة مستخدم.

5.البعد التعليمي والتقني

أبعد من كونه أداة للمستخدم النهائي، يُعد Calamares مورداً تعليمياً هاماً للمطورين. فقراءة ملفات إعداده وفهم كيفية تفاعل الوحدات مع النظام يكشف الكثير عن آلية عمل أنظمة لينكس نفسها. على سبيل المثال:

- **دراسة وحدة partition.confتُظهر كيف تتعامل الأنظمة مع مخططات الأقراص.**
- **وحدة users.confتكشف كيف يتم إنشاء الحسابات وضبط الصلاحيات.**
- **وحدة packages.confتسلط الضوء على كيفية دمج مدير الحزمpacman) ، apt، أو (dnf مع عملية التثبيت.**

لذلك، فإن التعمق في Calamares يمنح المطور فهماً مزدوجاً: كيف تُبنى التوزيعة من الداخل، وكيف تُقدّم للمستخدم في شكل واجهة سهلة الاستخدام.

6. Calamaresكجزء من فلسفة آرتش وحلوان لينكس

لطالما عُرف آرتش لينكس بفلسفته "افعلها بنفسك(Do It Yourself - DIY) "، حيث يُتوقع من المستخدم أن يضبط كل شيء يدوياً. ومع ذلك، يمكن أن تكون هذه الفلسفة حاجزاً أمام المستخدمين الجدد. هنا يأتي دور Calamares لحلوان لينكس: فهو يجمع بين مرونة آرتش وقوة الأدوات الرسومية الجاهزة، مما يوفر تجربة عصرية لا تستبعد المبتدئين وفي نفس الوقت لا تقيد المستخدمين المتقدمين.

.7محتوى الفصل

في الصفحات التالية، سنتناول:

- **الهيكل الداخلي لـ Calamares وملفاته الأساسية.**
- **كيفية تخصيص الواجهة البصرية والنصوص.**
- **أهم وحدات Calamares وكيفية إعدادها.**
- **دمج Calamares مع Archiso في بيئة حلوان لينكس.**

.8خاتمة المقدمة

فهم Calamares لا يقتصر على معرفة كيفية عمل مثبت رسومي؛ إنه نافذة على فلسفة تصميم التوزيعات الحديثة، حيث تتقاطع التكنولوجيا مع تجربة المستخدم، ويلتقي البعد التعليمي مع الجانب الجمالي. بالنسبة لحلوان لينكس، يمثل تبني وتخصيص Calamaresخطوة استراتيجية نحو تقديم توزيعة لا تقدم القوة التقنية فحسب، بل تسعى أيضاً لتقديم تجربة كاملة واحترافية تستحق التقدير.

3.2الهيكل الداخلي لـCalamares

.1نظرة عامة

بُني Calamares على مبدأ الوحدات .(modules) فكل وظيفة في عملية التثبيت (مثل تقسيم الأقراص، إنشاء المستخدمين، تثبيت الحزم) هي عبارة عن وحدة مستقلة يمكن تفعيلها، أو تعطيلها، أو تعديل ترتيبها.

.2المجلدات الرئيسية

- **:/etc/calamares/هذا هو الموقع الأساسي الذي يحتوي على ملفات الإعدادات.**
  - **:settings.confالملف الرئيسي الذي يحدد سير العمل) (workflow) تسلسل الوحدات.(**
  - **:/modulesكل وحدة لديها ملف إعدادات خاص بها، غالبًا ما يكون بامتداد) conf.مثل partition.conf، users.conf، .(packages.conf**

.3الملفات الأساسية

- **settings.conf:**
  - **يحدد أي الوحدات سيتم تحميلها وأيها سيتم تجاهلها.**
  - **يحتوي على أقسام مثل sequence، branding، و .welcome**
- **:/branding**

- يحتوي على الصور، الألوان، النصوص، وخيارات التصميم المخصصة للواجهة.

- **modules/:**
  - تشمل الوحدات الرئيسية:
    - **partition.conf:** للتحكم في طرق تقسيم الأقراص.
    - **users.conf:** لإعداد المستخدمين.
    - **packages.conf:** لاختيار الحزم.
    - **locale.conf:** لإعدادات اللغة والمنطقة.
    - **keyboard.conf:** لتخطيط لوحة المفاتيح.

4. خصائص الوحدات

يمكن تعريف كل وحدة بالخصائص التالية:

- **required:** أساسية ويجب تنفيذها.
- **optional:** يمكن تخطيها.
- **execOrder:** يحدد ترتيب التنفيذ (قبل/بعد وحدة معينة.)

5. آلية العمل

- يبدأ Calamares بتحميل settings.conf.
- يحدد الوحدات المطلوبة وفقًا للترتيب المحدد.
- تُطبّق كل وحدة على التوالي، مع تنفيذ أي سكربتات مضافة.
- أخيرًا، تُكتب السجلات إلى var/log/calamares.log.

يُقدم هذا القسم تفصيلاً واضحًا للهيكل الداخلي لـCalamares ، مع التركيز على تصميمه النمطي وأدوار ملفات ومجلدات الإعدادات الرئيسية.

3.3 تخصيص واجهة Calamares (Branding & UI Customization)

1. فلسفة وأهمية التخصيص

من بين كل العناصر التي تشكّل تجربة أي توزيعة لينكس، يظل المثبت هو الواجهة الأولى التي يواجهها المستخدم. هذه اللحظة ليست تقنية فقط، بل نفسية أيضًا. فالانطباع الذي يتركه المثبت يمكن أن يحدد مصير العلاقة بين المستخدم والنظام: هل سيرى التوزيعة كمشروع احترافي وجاد، أم مجرد تجميعة من هواة تفتقر للهوية؟

الهوية البصرية كرسالة غير منطوقة

الهوية البصرية ليست مجرد ألوان أو شعارات؛ إنها رسالة تترجم فلسفة التوزيعة إلى شكل ملموس. عندما يفتح المستخدم نافذة التثبيت ويرى اسم التوزيعة، شعارها، ألوانها المتناسقة، وحتى الخلفية المميزة خلف النوافذ، فإنه يدرك على الفور "روح المشروع ." Calamaresيمنح المطور هذه القدرة: على أن يضفي على كل جزء من واجهة التثبيت معنى ورسالة.

بين البساطة والتعقيد

عبر تاريخ لينكس، انقسمت التوزيعات إلى مدرستين فكريتين:

- **مدرسة البساطة الجذرية :ويمثلها آرتش لينكس نفسه، حيث يُتوقع من المستخدم كتابة الأوامر وإعداد النظام من الصفر.**
- **مدرسة السهولة البصرية :مثل أوبونتو ومانجارو، التي تقدم تجربة رسومية كاملة منذ لحظة التثبيت.**

لكن السؤال الذي يطرحه مطورو التوزيعات الحديثة هو: هل يمكن دمج قوة المدرسة الأولى مع أناقة الثانية؟ هنا يكمن دور Calamares، وهنا يبرز دور "العلامة التجارية (Branding) "كحلقة وصل بين فلسفة المشروع وتجربة المستخدم.

حلوان لينكس كدراسة حالة

توزيعة حلوان لينكس، المبنية على آرتش، لم تتوقف عند مجرد إعادة تجميع لأدوات .Archiso لقد سعت لتقديم هوية متكاملة: شعار واضح، ألوان متناسقة، ورسالة جادة بأن هذه التوزيعة هي مشروع مكتمل. لم يكن اختيار Calamares صدفة، بل قرارًا استراتيجيًا: "إذا أردنا أن نقنع المستخدم الغربي والعربي على حد سواء بأن حلوان لينكس هو مشروع ذو قيمة، فيجب أن نبدأ من حيث يبدؤون: من نافذة التثبيت."

لماذا تُعد العلامة التجارية مهمة؟

- **تبني الثقة :المثبت الاحترافي يوحي بأن فريقًا منظمًا يقف خلف المشروع.**
- **تميز التوزيعة :تمنع المثبت من الظهور كنسخة مباشرة من أي توزيعة أخرى.**
- **تساعد على التبني :المستخدم الجديد قد لا يقرأ التوثيق لكنه سيتأثر بصريًا على الفور.**
- **توفر تجربة موحدة :الهوية البصرية الممتدة من موقع التوزيعة إلى شاشة التثبيت تخلق "سردًا واحدًا."**

.2ملفات الإعدادات الأساسية فيCalamares

تستند قوة Calamares على فكرة أن كل جانب تقريبًا من جوانب عملية التثبيت يمكن تخصيصه. يتراوح هذا من ترتيب الخطوات التي يتبعها المستخدم إلى الألوان، والشعارات، وحتى نصوص الترحيب. تأتي هذه القدرة على التخصيص من مجموعة من الملفات النصية الموجودة في مجلد ./etc/calamares/

أهم هذه الملفات هي:

- **:settings.confالملف الرئيسي الذي يحدد ترتيب الوحدات ويربط المثبت بالهوية البصرية المختارة.**
- **:branding.descالملف المسؤول عن تعريف الهوية البصرية (الألوان، الشعارات، الروابط، حجم النافذة.(**
- **:stylesheet.qssملف أوراق أنماط (stylesheet) يشبه CSS يسمح بالتحكم الدقيق في مظهر الواجهة.**

في هذا القسم، سنشرح كل ملف بالتفصيل مع أمثلة عملية.

2.1ملفsettings.conf

هذا هو الملف المركزي في .Calamares عند إطلاق المثبت، يتم قراءة settings.confأولاً لتحديد:

- **أي الوحدات يجب تحميلها.**
- **بأي ترتيب يجب عرضها أو تنفيذها.**
- **أي هوية بصرية (branding) يجب استخدامها.**

مثال عملي من حلوان لينكس

المقتطف التالي مأخوذ مباشرة من ملف settings.confالخاص بتوزيعة حلوان لينكس:

```
branding:

helwan

sequence:

- show: welcome

- show: locale

- show: keyboard

- show: partition

- show: users

- show: summary

- exec: unpackfs

- exec: packages

- exec: bootloader

- exec: shellprocess@before

- exec: shellprocess@final
```

تحليل:

- **branding: helwan:هذا السطر يوجه Calamares لاستخدام الهوية البصرية المسماة helwan، الموجودة في مجلد branding/helwan/.**
- **sequence::هذا القسم يحدد تسلسل خطوات التثبيت.**
  - **show:تستخدم لعرض وحدات واجهة المستخدم (UI) التي تتطلب تفاعلًا من المستخدم (مثل شاشات الترحيب، واختيار اللغة، وتقسيم القرص.)**

- **exec:** تستخدم لتنفيذ الوحدات في الخلفية دون الحاجة إلى تدخل المستخدم (مثل فك ضغط نظام الملفات الأساسي unpackfs، وتثبيت الحزم، وتثبيت أداة الإقلاع .(bootloader

- **shellprocess@before:** هذا يعرض سكربتات تُشغّل قبل الخطوات النهائية.
- **shellprocess@final:** هذا يعرض سكربتات تُشغّل في النهاية، غالبًا للتنظيف أو إكمال المهام النهائية.

بالتأكيد، إليك الترجمة الكاملة والدقيقة للنص الذي أرسلته.

أهمية ملف :settings.conf

- **إنه يحدد تجربة المستخدم بأكملها من البداية إلى النهاية.**
- **يمكن اعتباره "خريطة الطريق" لرحلة التثبيت.**
- **أي تغيير فيه يمكن أن يعيد تشكيل الانطباع الأول للتوزيعة بالكامل.**

2.2 ملف branding.desc

إذا كان settings.conf هو خريطة الطريق، فإن branding.desc هو هوية الرحلة. هذا الملف يحدد ما سيراه المستخدم: الألوان، والشعارات، والنصوص، وحجم النافذة.

مقتطف من حلوان لينكس:

```
componentName:   helwan

welcomeStyleCalamares:   false

windowSize: 1200px,700px

windowPlacement: center

sidebar: widget

navigation: widget

strings:

   productName:         Helwan Linux

   version:             2024-12

   bootloaderEntryName: Helwan Linux

images:

    productBanner:      "banner.png"

   productIcon:         "install.png"

   productLogo:         "logo.png"
```

```
    productWallpaper:    "wallpaper.png"
    productWelcome:      "icon.png"
style:
    SidebarBackground:       "#56728b"
    SidebarText:             "#FFFFFF"
    SidebarTextCurrent:      "#292F34"
    SidebarBackgroundCurrent:  "#00CED1"
slideshow: "show.qml"
slideshowAPI: 2
```

تحليل مفصل:

- **componentName: يحدد اسم مجلد العلامة التجارية.(helwan)**
- **welcomeStyleCalamares: إذا كانت القيمة true، فسيتم عرض نص "Welcome to the Calamares installer for Helwan Linux"؛ إذا كانت false، فسيتم عرض."Welcome to the Helwan Linux installer"**
- **windowSize & windowPlacement: حجم نافذة Calamares (1200x700) وموضعها (في المنتصف.)**
- **sidebar & navigation: تحدد نوع اللوحات الجانبية والتنقل.(widget, qml, none)**
- **strings: يحتوي على اسم المنتج، ورقم الإصدار، والاسم الذي يظهر في أداة الإقلاع.(GRUB)**
- **images: يحدد الشعارات والخلفيات المستخدمة.**
- **style: يحدد ألوان الواجهة (خلفية الشريط الجانبي، لون النص.(...**
- **slideshow: ملف QML يتم عرضه أثناء التثبيت لتوفير تجربة مستخدم غنية.**

لماذا يُعد branding.desc مهمًا؟

- **إنه الملف الذي يمنح التوزيعة هويتها البصرية المميزة.**
- **أي تعديلات عليه تترجم مباشرة إلى واجهة مختلفة تمامًا.**
- **يسمح بتجربة موحدة من شاشة الترحيب الأولى إلى شاشة الإكمال النهائية.**

2.3 ملف stylesheet.qss

بينما يوفر branding.desc تخصيصًا عامًا (الألوان، الصور، الشعارات)، يوفر stylesheet.qss تخصيصًا دقيقًا على مستوى المكون.(widget)

مقتطف من حلوان لينكس:

CSS

```
#debugButton
{
   background-color: #31363b;
   font: 10px;
   color: #ffffff;
   height: 32px;
   padding: 5px;
   border: none;
}
#debugButton:pressed
{
   color: #00aeff;
}
#aboutButton
{
   background-color: #31363b;
   font: 10px;
   color: #ffffff;
   height: 32px;
   padding: 5px;
   border: none;
}
#aboutButton:hover
{
    color: #00aeff;
}
```

تحليل:

- **تُستخدم المعرفات (IDs) مثل #debugButton أو #aboutButton للتحكم في ألوان وأحجام الأزرار.**
- **يمكن تغيير حالة الزر عند الضغط عليه (:pressed) أو عند مرور مؤشر الفأرة فوقه.(:hover)**
- **نفس مفهوم CSS في تطوير الويب يُطبّق هنا على واجهة Qt.**

أهمية stylesheet.qss:

- **يمنح المطور حرية كاملة لتخصيص كل تفصيلة في الواجهة.**
- **يسمح بتجربة مستخدم متسقة مع بقية النظام (مظهر موحد.)**
- **يعكس جدية التوزيعة (مظهر احترافي مقابل مظهر بسيط وافتراضي.)**

3.4آلية عمل ملف show.qml

يُبنى عرض الشرائح (slideshow) باستخدام QML لغة واجهة رسومية هي جزء من إطار عمل Qt. الملف الذي قدمته هو Presentation يحتوي على مجموعة من كائنات Slide، يعرض كل منها صورة (مثل slide1.png، slide2.png، إلخ.)

مقتطف مبسط:

```
QML

Presentation

{

  Timer {

    id: advanceTimer

    interval: 15000

    running: presentation.activatedInCalamares

    repeat: true

    onTriggered: nextSlide()

  }


  Slide {

    Image {

        source: "slide1.png"
```

```
        anchors.fill: parent
        fillMode: Image.Stretch
      }
    }
}
```

تحليل:

- **Presentation: العنصر الحاوي الرئيسي للشرائح.**
- **Timer: مؤقت يحدد متى يتم الانتقال إلى الشريحة التالية (هنا، كل 15 ثانية).**
- **Slide: يمثل شريحة واحدة.**
- **Image: الصورة المعروضة داخل الشريحة.**

3.4.1 هيكل عرض الشرائح في حلوان لينكس

يحتوي ملف show.qml على 23 شريحة (slide1.png إلى slide23.png). كل شريحة هي صورة منفصلة تُعرض بالتسلسل، بمعدل 15 ثانية لكل شريحة.

وهذا يوفر تجربة غنية:

- **يمكن استخدام الشرائح الأولى لتقديم التوزيعة (التاريخ، الفلسفة).**
- **يمكن للشرائح الوسطى عرض صور لسطح المكتب والتطبيقات المثبتة مسبقًا.**
- **يمكن للشرائح النهائية توجيه المستخدم إلى المجتمع، أو الدعم، أو الموارد التعليمية.**

3.4.2 إصدارات API V1 و V2

يدعم Calamares طريقتين لإدارة عرض الشرائح:

- **API V1: يقوم بتشغيل العرض مرة واحدة ويستمر حتى اكتمال التثبيت.**
- **API V2: يوفر مرونة أكبر عبر دوال مثل onActivate() و onLeave().**

في ملف حلوان لينكس، نجد:

```
QML
function onActivate() {
   console.log("QML Component activated");
   presentation.currentSlide = 0;
}
```

```
function onLeave() {
   console.log("QML Component deactivated");
}
```

هذا يعني أنه بمجرد دخول المستخدم إلى واجهة عرض الشرائح، يبدأ العرض من الشريحة الأولى، ويتوقف بمجرد مغادرتها.

3.4.3تخصيص الشرائح

- **الصور :يمكنك استبدال slide1.pngبصورة مخصصة، على سبيل المثال، شعار حلوان لينكس على خلفية داكنة.**
- **النص :يمكنك إضافة عناصر Textفوق الصور لإعلام المستخدم بالميزة المعروضة.**
- **التفاعل :باستخدامQML ، يمكنك إضافة Animationsأو حتى مقطع فيديو قصير.**

مثال على إضافة نص فوق صورة:

```
QML
Slide
{
   Image { source: "slide1.png"; anchors.fill: parent }
   Text {
      text: "Welcome to Helwan Linux"
      color: "white"
      font.pixelSize: 32
      anchors.centerIn: parent
   }
}
```

3.4.5أفضل الممارسات لتصميم عرض الشرائح

- **وضوح الرسالة :يجب أن تنقل كل شريحة فكرة واحدة (مثلاً: "الأمان"، "السرعة"، "المجتمع.("**
- **اتساق الألوان :يجب أن تعكس الشرائح العلامة التجارية للتوزيعة.**
- **نصوص مقتضبة :لن يقرأ المستخدم فقرات طويلة أثناء الانتظار.**
- **استخدام لقطات شاشة فعلية :صور سطح المكتب أو التطبيقات المثبتة مسبقًا أكثر إقناعًا من الشعارات المجردة.**

- **مراعاة الوقت :إذا كانت كل شريحة 15 ثانية ولديك 20 شريحة، فإن العرض الإجمالي هو 5 دقائق، وهو مناسب لمتوسط وقت التثبيت.**

3.4.6حلوان لينكس كدراسة حالة

يمثل عرض الشرائح في حلوان لينكس أداة أساسية للتواصل مع المستخدم أثناء عملية التثبيت. فبينما يقوم النظام بنسخ الملفات وإعداد القرص الصلب، يمكن لفريق التطوير أن يقدم رسالة واضحة ومنظمة عن هوية التوزيعة وميزاتها. وبهذه الطريقة، يتحول وقت الانتظار — الذي قد يكون مملًا في الأنظمة الأخرى — إلى تجربة تعليمية وبصرية شاملة.

المرحلة الأولى: تقديم المشروع وفلسفته (الشرائح 1-5 (تبدأ الشرائح الأولى بتقديم موجز لـ حلوان لينكس. يمكن أن تتضمن هذه الشرائح:

- **صورة مع الشعار الرسمي للتوزيعة وعبارة ترحيب مثل: "أهلاً بك في حلوان لينكس، رؤية جديدة لتوزيعة مبنية على آرتش."**
- **شريحة أخرى تشرح فلسفة التوزيعة: البساطة، المرونة، والالتزام بالشفافية.**
- **صور توضح العلاقة بين حلوان لينكس وآرتش لينكس، مصحوبة بعبارة مثل: "قوة آرتش، مع سهولة أكبر ."تمنح هذه المرحلة المستخدم انطباعًا أوليًا بأن التوزيعة ليست مجرد مشروع عشوائي بل رؤية متكاملة ذات أهداف واضحة.**

المرحلة الثانية: عرض واجهة سطح المكتب والتطبيقات الافتراضية (الشرائح 6-10 (تركز هذه المجموعة من الشرائح على الجوانب العملية التي سيجدها المستخدم فورًا بعد التثبيت:

- **لقطات شاشة فعلية لسطح المكتب الافتراضي، مع إبراز بساطة التصميم وأناقة الهوية البصرية.**
- **عرض بعض التطبيقات الأساسية المثبتة مسبقًا مثل مدير الملفات، أو مشغل الوسائط، أو مركز البرامج.**
- **التأكيد على سهولة الاستخدام: قوائم واضحة، أيقونات مميزة، وتجربة متكاملة تسمح للمستخدم بالبدء في العمل فورًا بعد التثبيت دون إعدادات معقدة.**

المرحلة الثالثة: إبراز الميزات الفريدة للتوزيعة (الشرائح 11-15 (بعد أن يتعرف المستخدم على الواجهة، يحين وقت تسليط الضوء على ما يميز حلوان لينكس عن غيره:

- **شرائح تشرح التركيز على الأمان: مثل دعم التشفير (LUKS) أو التحديثات السريعة.**
- **عرض الأداء العالي عبر أمثلة: أوقات تشغيل سريعة، استهلاك منخفض للموارد.**
- **الحديث عن الأدوات المدمجة: مثل السكربتات التي تبسط إدارة النظام أو البرامج التي تم تطويرها خصيصًا لحلوان .يجب أن تحمل كل شريحة في هذه المرحلة رسالة قصيرة ومباشرة، مثل: "حلوان لينكس يحمي بياناتك"، "أداء يلبي توقعاتك"، "أدواتك في متناول يدك."**

المرحلة الرابعة: المجتمع، الدعم، والموارد (الشرائح 16-23 (تهدف المرحلة النهائية من العرض إلى بناء علاقة طويلة الأمد مع المستخدم:

- **شريحة تحتوي على روابط للموقع الرسمي.**
- **أخرى تعرض GitHub أو مستودع المشروع المفتوح المصدر، لتأكيد الشفافية وتشجيع المساهمات.**

- شريحة لصفحة الدعم ومنتدى المستخدمين.
- يمكن إضافة شريحة أخيرة برسالة ملهمة: "حلوان لينكس ليس مجرد نظام، بل مجتمع يرحب بك."

ملخص التجربة بهذا الترتيب، تتحول الشرائح من مجرد صور إلى رحلة شاملة:

- البداية :مقدمة للفلسفة.
- الوسط :عرض للواقع العملي والميزات.
- النهاية :فتح الأبواب للمستخدم للانضمام إلى المجتمع .هذا النهج يجعل حلوان لينكس يظهر كتوزيعة ناضجة ويمنح المستخدم أسبابًا واضحة للبقاء معها بعد التثبيت. في النهاية، يصبح عرض الشرائح أكثر من مجرد عنصر تجميلي: إنه أداة استراتيجية للتسويق والتعليم وبناء الثقة.

3.5إدارة الأقراص عبر partition.confفيCalamares

3.5.1مقدمة

تقسيم القرص (Disk Partitioning) هو الخطوة الأكثر أهمية وجوهرية في عملية تثبيت أي نظام تشغيل. فهي التي تحدد مكان تخزين النظام، وأنواع الملفات، وما إذا كانت البيانات الموجودة سيتم الحفاظ عليها أو مسحها بالكامل.

في أنظمة لينكس، تصبح هذه العملية أكثر تعقيدًا بسبب تنوع أنظمة الملفات (ext4, btrfs, xfs...) وأنواع أقسام الإقلاع المختلفة (EFI vs. BIOS)، إلى جانب خيارات التشفير وملفات المبادلة.(swap files)

لهذا السبب، يقدم Calamares وحدة مستقلة مسؤولة عن إدارة الأقراص.partition.conf :

تسمح هذه الوحدة لمطوري التوزيعة بإعداد جميع الإعدادات الافتراضية مسبقًا، مع منح المستخدمين حرية التخصيص.

في حالة حلوان لينكس، هذه الخطوة حاسمة لأنها تمثل الخط الفاصل بين:

- تجربة واضحة، بسيطة، واحترافية للمستخدم المبتدئ.
- تجربة مربكة قد تؤدي إلى فقدان البيانات أو نفور المستخدم من النظام.

3.5.2الهيكل العام لملف partition.conf

الملف مكتوب بصيغة YAMLويحتوي على عدة مفاتيح تتحكم في تفاصيل دقيقة للغاية. أهم هذه المفاتيح تشمل:

- efiSystemPartition:يحدد المسار لقسم) EFI شائعًا .(/boot/efi
- efiSystemPartitionSize:الحجم الافتراضي للقسم (مثل .(300MiB
- userSwapChoices:يحدد خيارات المبادلة المتاحة.(none, small, suspend, file)
- defaultFileSystemType:يضبط نظام الملفات الافتراضي (مثل .(ext4
- availableFileSystemTypes:قائمة بأنظمة الملفات المتاحة.

- **partitionLayout: مخطط تقسيم محدد مسبقًا يتم تطبيقه تلقائيًا.**
- **drawNestedPartitions: يعرض الأقسام المنطقية بشكل صحيح.**
- **initialPartitioningChoice: الخيار الافتراضي عند ظهور واجهة التقسيم = none) أمان للمستخدم.(**
- **enableLuksAutomatedPartitioning: يمكّن التشفير التلقائي لـ LUKS.**

3.5.3 تحليل مفصل للإعدادات

3.5.3.1 قسم نظام EFI

- **المسار الافتراضي /boot/efi :**
- **الحجم) 300MiB :قابل للتعديل(**
- **الدور :أساسي لأنظمة UEFI الحديثة، ويحتوي على ملفات أداة الإقلاع GRUB) أو systemd-boot).**
- **مشكلة شائعة :إذا لم يتم إنشاء هذا القسم، فإن النظام لن يعمل أبدًا.**

3.5.3.2 خيارات المبادلة (Swap)

هذا القسم أو الملف مسؤول عن إدارة الذاكرة الافتراضية. الخيارات هي:

- **none: لا يوجد مبادلة.**
- **small: يصل إلى 8GiB أو 10% من القرص.**
- **suspend: يساوي حجم الذاكرة العشوائية) (RAM مطلوب لوظيفة .(Hibernate**
- **file: ملف مبادلة يقع داخل قسم الجذر (الأبسط والأسهل.(**

مثال :في حلوان لينكس، يمكن ضبط الخيار الافتراضي على file لسهولة الإدارة.

3.5.3.3 أنواع نظام الملفات

- **الافتراضي) ext4 :مستقر ومُختبر جيدًا.(**
- **خيارات إضافية) btrfs :للقطات، (Timeshift،) xfs للخوادم(،) f2fs لأقراص SSD/Flash).**

توصية حلوان لينكس:

- **ext4 = الافتراضي.**
- **btrfs = خيار للمستخدمين المتقدمين.**

3.5.3.4تخطيط الأقسام(Partition Layout)

هذا قسم متقدم يسمح بتعريف مخطط تقسيم كامل مسبقًا:

```
YAML

partitionLayout:

 - name: "rootfs"

   filesystem: "ext4"

   mountPoint: "/"

   size: 30G

 - name: "home"

   filesystem: "ext4"

   mountPoint: "/home"

   size: 100%
```

في حلوان لينكس، يمكن تقديم هذا الخيار كخيار "تثبيت تلقائي(Auto Install) "، مما يوفر للمستخدم:

- **EFI (300MiB)**
- **الجذر(30GiB) (Root)**
- **) Homeالباقي(**
- **ملف مبادلة) (Swap file) بناءً على الذاكرة العشوائية(RAM**

3.5.3.4أنواع جدول الأقسام(Partition Table Types)

- **:GPTالأنسب للأجهزة الحديثة.(UEFI)**
- **:MSDOS (MBR)أقدم ومحدود (أربعة أقسام أساسية فقط.(**
- **توصية :اجعل GPT الخيار الافتراضي في حلوان لينكس.**

3.5.3.5تشفيرLUKS

- **تفعيل التشفير يضيف مستوى عالٍ جدًا من الأمان.**
- **تحذير :Calamares لا تقم بتعطيله إلا لوجود سبب قوي.**
- **في حلوان لينكس:**
  - **اجعل خيار LUKS متاحًا، لكن ليس مفعلًا افتراضيًا.**
  - **وثّق الخطوات للمستخدمين المتقدمين.**

3.5.4سير العمل

- عند بدءCalamares ، يقرأ ملف .partition.conf
- يحدد أنواع التقسيم المدعومة بناءً على إعدادات التوزيعة.
- يقدم واجهة للمستخدم تحتوي على خيارات (مسح، استبدال، بجانب، يدوي.(
- إذا تم اختيار "تلقائي"، فإنه يستخدم partitionLayoutأو .defaultFileSystemType
- يكتب التغييرات على القرص باستخدام مكتبة .KPMCore
- يسجل النتائج في .var/log/calamares.log

3.5.5حلوان لينكس كدراسة حالة

الإعدادات المقترحة:

- efiSystemPartition: /boot/efiبحجم MiB.300
- swap: fileكخيار افتراضي.
- defaultFileSystemType: ext4.
- availableFileSystemTypes: [ext4, btrfs, xfs].
- partitionLayout: Root (30GiB), Home (الباقي, EFI, (ملف.Swap)
- initialPartitioningChoice: noneيجبر المستخدم على الاختيار يدويًا.( )
- enableLuksAutomatedPartitioning: trueلكنه ليس الخيار الافتراضي.( )

الفوائد:

- سهل للمبتدئين (التقسيم التلقائي.(
- مرن للخبراء (التقسيم اليدوي.(Btrfs +
- يعكس احترافية حلوان لينكس.

3.5.6المشاكل الشائعة والحلول

- عدم العثور على قسم :EFI أضف /boot/efiيدويًا.
- المبادلة صغيرة جدًا :اختر suspendبدلاً من .small
- تعارض :GPT vs MBR قم بتمكين GPT كافتراضي.
- المستخدم يحذف الجذر عن طريق الخطأ :نفّذ تحذيرًا إضافيًا.

3.5.7أفضل الممارسات

- لا تجعل خيار "مسح" هو الخيار الافتراضي.
- قدّم خيار "التقسيم التلقائي" بمخطط واضح.(Root + Home + EFI)
- اعرض btrfsكخيار، وليس كافتراضي.
- وثّق كل هذا في دليل حلوان لينكس + عرض الشرائح.

3.5.8خاتمة

ملف partition.confليس مجرد إعداد في الكواليس؛ إنه حجر الزاوية في تجربة تثبيت ناجحة. من خلال التحكم الدقيق فيه، يمكن لحلوان لينكس أن يقدم تجربة آمنة، واحترافية، ومرنة. فإعداد تقسيم القرص بشكل صحيح يعكس صورة التوزيعة بأكملها ويغرس الثقة في المستخدم بأنه يستخدم نظامًا مصممًا بعناية، وليس مجرد تجميعة عشوائية.

3.6وحدة users.conf

3.6.1مقدمة

تُعد إدارة المستخدمين والحسابات واحدة من أكثر الخطوات حساسية وأهمية في تثبيت أي نظام تشغيل. على عكس التقسيم—الذي يتعامل مع التخزين—تحدد هذه الخطوة كيفية تفاعل المستخدم البشري مع النظام.

وهي تغطي العناصر الحاسمة التالية:

- إنشاء حساب مستخدم أساسي.
- حالة حساب) rootالمدير.(
- المجموعات التي تحدد الصلاحيات والوصول.
- سياسات الأمان، خاصة قوة كلمة المرور.

بالنسبة لتوزيعة مثل حلوان لينكس، فإن وحدة users.confلا تتعلق فقط بالتفاصيل التقنية؛ بل تتعلق بتحقيق توازن بين سهولة استخدام المستخدم وأمان النظام.

3.6.2الهيكل العام لملف users.conf

ملف users.confمكتوب بصيغة YAML ويتكون من عدة مفاتيح تحدد الإعدادات الافتراضية لإنشاء المستخدم. من أهمها:

- defaultGroups:المجموعات التي سينتمي إليها المستخدم افتراضيًا.
- autologinGroup & doAutologin:إعدادات تسجيل الدخول التلقائي.
- sudoersGroup:يحدد صلاحيات sudo.
- setRootPassword & doReusePassword:يحدد سلوك حساب root.

- **passwordRequirements: يفرض تعقيد كلمة المرور.**
- **user: يتحكم في الصدفة (shell) الافتراضية وأسماء تسجيل الدخول الممنوعة.**
- **hostname: يحدد كيفية ضبط اسم مضيف الجهاز.**

بالإضافة إلى ذلك، يخزن Calamares مفاتيح التخزين العامة (Global Storage) مثل hostname وusername وpassword بناءً على إدخال المستخدم أثناء التثبيت.

3.6.3 المجموعات الافتراضية (Default Groups)

تحدد قائمة defaultGroups المجموعات التي ينضم إليها المستخدم الجديد تلقائيًا. وهذا يحدد الوصول إلى الأجهزة وميزات النظام.

مثال من حلوان لينكس:

YAML

```
defaultGroups:
  - name: users
    must_exist: true
    system: true
  - lp
  - video
  - network
  - storage
  - name: wheel
    must_exist: false
    system: true
  - audio
```

نقاط رئيسية:

- **users: المجموعة الأساسية لجميع الحسابات غير النظامية.**
- **video: مطلوب للوصول إلى وحدة معالجة الرسوميات (GPU) والعرض.**
- **network: صلاحية إعداد الشبكات.**
- **storage: الوصول إلى وسائط التخزين القابلة للإزالة (محركات أقراص USB).**
- **wheel: مجموعة إدارية خاصة لصلاحيات sudo.**

- **audio: الوصول إلى الميكروفون وأجهزة الصوت.**

في حلوان لينكس :تضمن هذه المجموعة من المجموعات أن المستخدم الجديد لديه جميع الصلاحيات الضرورية دون إعدادات إضافية.

3.6.4 تسجيل الدخول التلقائي(Autologin)

يسمح Calamares للتوزيعات بتهيئة تسجيل الدخول التلقائي:

```
autologinGroup: auto

doAutologin: true
```

- **autologinGroup: يحدد مجموعة مسؤولة عن تسجيل الدخول التلقائي.**
- **doAutologin: يحدد ما إذا كان تسجيل الدخول التلقائي ممكّنًا افتراضيًا.**

الإيجابيات والسلبيات:

- **سهولة الاستخدام :للمبتدئين والأنظمة الحية.(live systems)**
- **أمان منخفض :حيث يمكن لأي شخص لديه وصول مادي تسجيل الدخول.**

توصية لحلوان لينكس:

- **مكّن تسجيل الدخول التلقائي في نظام ISO الحي فقط.**
- **عطّله في الأنظمة المثبتة لإعطاء الأولوية للأمان.**

3.6.5 إعداد Sudoers

```
sudoersGroup: wheel

sudoersConfigureWithGroup: true
```

يضمن هذا الإعداد أن أي مستخدم يُضاف إلى مجموعة wheel يتمتع بامتيازات إدارية. ينشئ Calamares تلقائيًا ملف /etc/sudoers.d/10-installer لمنح الصلاحيات.

الآثار المترتبة:

- **wheel: مجموعة قياسية عبر العديد من توزيعات لينكس.**
- **sudoersConfigureWithGroup: true: يضمن التوافق مع إعدادات sudo المتقدمة.**

في حلوان لينكس:

- **قم دائمًا بإضافة المستخدم الأول إلى مجموعة .wheel**
- **وثّق هذا بوضوح في دليل التثبيت.**

3.6.6 إعدادات حساب Root

سلوك حساب rootأمر بالغ الأهمية.

setRootPassword: true

doReusePassword: true

- **setRootPassword: يمكّن كلمة مرور منفصلة لـ root.**
- **doReusePassword: يعيد استخدام كلمة مرور المستخدم لـ root.**

السيناريوهات:

- **Rootمعطل :أكثر أمانًا للمبتدئين؛ يعتمد على sudo.**
- **Rootممكّن بنفس كلمة المرور :مريح ولكنه أقل أمانًا.**
- **Rootممكّن بكلمة مرور مختلفة :الأفضل للمستخدمين المتقدمين.**

الإعداد الافتراضي لحلوان لينكس:

- **عطّل تسجيل دخول) rootأفضل ممارسات الأمان الحديثة.(**
- **استخدم sudoللإدارة.**
- **وفّر خيارًا للمستخدمين المتقدمين لتمكين root.**

3.6.7سياسة كلمة المرور

تؤثر قوة كلمة المرور مباشرة على أمان النظام. تشمل الخيارات:

- **passwordRequirements: يفرض حدًا أدنى للطول والتعقيد.**
- **allowWeakPasswords: ما إذا كان يُسمح بكلمات المرور الضعيفة.**
- **allowWeakPasswordsDefault: ما إذا كان مربع الاختيار ممكّنًا افتراضيًا.**

مثال:

allowWeakPasswords: false

allowWeakPasswordsDefault: true

أفضل الممارسات:

- **ارفض كلمات المرور الفارغة أو التافهة.**
- **اطلب 8 أحرف على الأقل.**
- **استخدم libpwqualityعند توفرها.**

سياسة حلوان لينكس:

- **فرض كلمات مرور قوية بشكل افتراضي.**
- **السماح بكلمات المرور الضعيفة فقط إذا تجاوز المستخدم هذا الخيار صراحةً.**

3.6.8صدفة المستخدم (User Shell) والأسماء الممنوعة

```
user:
  shell: /bin/bash
  forbidden_names: [ root ]
```

- **shell:يحدد الصدفة الافتراضية.(bash, zsh, fish)**
- **forbidden_names:يمنع استخدام الأسماء المحجوزة للنظام.**

الإعداد الافتراضي لحلوان لينكس:

- **الصدفة الافتراضية.bash/bin/ :**
- **وفّر خيارًا اختياريًا لـ zshللمستخدمين المتقدمين.**

3.6.9تهيئة اسم المضيف(Hostname)

```
hostname:
  location: EtcFile
  writeHostsFile: true
  forbidden_names: [ localhost ]
```

- **location:حيث يتم تخزين اسم المضيف.(/etc/hostname)**
- **writeHostsFile:يحدّث ملف .etc/hosts/**
- **template:يمكن أن ينشئ اسم المضيف تلقائيًا من معلومات المستخدم/المنتج.**

مثال على القالب:

```
template: "${login}-helwan"
```

إذا كان اسم تسجيل الدخول هو ahmed، يصبح اسم المضيف .ahmed-helwan

حلوان لينكس:

- **قالب بسيط ولكن يحمل علامة تجارية لاسم المضيف.**
- **يتجنب التعارضات مع .localhost**
-

3.6.10حلوان لينكس كدراسة حالة

الإعدادات الافتراضية المقترحة:

- **المجموعات.users, audio, video, network, wheel :**
- **تسجيل الدخول التلقائي :فقط في نظام ISO الحي.**
- **:Rootمعطل افتراضيًا، sudoممكّن.**
- **كلمات المرور :سياسة قوية مفروضة.**
- **الصدفة bash :(Shell) افتراضيًا، zshاختياري.**
- **اسم المضيف.helwan-${login} :**

هذا المزيج يضمن:

- **الأمان للمستخدمين المتقدمين.**
- **البساطة للمبتدئين.**
- **هوية مميزة لحلوان لينكس.**

3.6.11أفضل الممارسات

- **لا تمكّن تسجيل الدخول التلقائي افتراضيًا.**
- **افصل بوضوح بين الإعدادات الافتراضية للمبتدئين والخيارات المتقدمة.**
- **قم دائمًا بفرض سياسة كلمات مرور قوية.**
- **وثّق سياسة sudoفي دليل التثبيت.**
- **وفّر أصدافًا حديثة (zsh, fish) للمستخدمين المحترفين.**

3.6.12خاتمة

ملف users.confهو أكثر من مجرد ملف إعدادات؛ إنه قلب تجربة المستخدم. فهو يحدد كيفية تفاعل المستخدم مع النظام، ومدى أمان التثبيت، ومدى احترافية التوزيعة.

بالنسبة لحلوان لينكس، فإن ضبط هذه الإعدادات بعناية يضمن توازنًا بين قابلية الاستخدام والأمان، مع عكس فلسفة المشروع المتمثلة في الوضوح، والمرونة، والاحترافية.

3.7فك ضغط نظام الملفات باستخدام unpackfs.conf

3.7.1مقدمة

بعد تقسيم القرص وإعداد المستخدم(users.conf) ، تأتي المرحلة الأكثر عملية: نسخ ملفات النظام إلى القرص الصلب.

في توزيعات لينكس المبنية على آرتش (مثل حلوان لينكس)، تُحصر ملفات النظام الأساسية عادةً داخل صورة مضغوطة (عادةً SquashFS).وتُعد وحدة unpackfsفي Calamares مسؤولة عن استخراج هذه الملفات إلى القرص المستهدف.

بدون هذه الخطوة، يكون التثبيت مجرد إعداد بيئي؛ ولكن مع unpackfs.conf، يتحول النظام من فكرة إلى حقيقة.

3.7.2الهيكل العام لملف unpackfs.conf

الملف مكتوب بصيغة YAML ويحتوي على قائمة فك ضغط ,(unpack) حيث يمثل كل عنصر عملية نسخ أو استخراج.

العناصر الأساسية:

- source:موقع الملف أو الصورة داخل النظام الحي.(live system)
- sourcefs:نوع المصدر.(squashfs, ext4, file)
- destination:الموقع المستهدف داخل النظام الجديد (نسبيًا إلى .(rootMountPoint

خيارات إضافية:

- exclude:يستبعد ملفات معينة أثناء النسخ.
- excludeFile:ملف خارجي يحتوي على قائمة بالاستثناءات.
- weight:يُستخدم للتحكم في شريط التقدم عندما يكون هناك عدة عمليات.

3.7.3مثال من حلوان لينكس

في حلوان لينكس، يحتوي الملف على عمليتين رئيسيتين:

```
unpack:
  - source: "/run/archiso/bootmnt/arch/x86_64/airootfs.sfs"
    sourcefs: "squashfs"
    destination: ""
  - source: "/run/archiso/bootmnt/arch/boot/x86_64/vmlinuz-linux"
    sourcefs: "file"
    destination: "/boot/vmlinuz-linux"
```

تحليل:

- السطر الأول يفك ضغط صورة SquashFS (airootfs.sfs) إلى المجلد الجذر للنظام.(/)

- السطر الثاني ينسخ نواة النظام (vmlinuz-linux) من ملف ISO إلى .boot/

بهذه الخطوات، يصبح نظام الجذر جاهزًا تقريبًا، في انتظار إعداد fstabوتهيئة أداة الإقلاع.

3.7.4أنواع المصادر المدعومة

- :squashfsالأكثر شيوعًا، حيث تعتمد عليه جميع الأنظمة الحية لضغط ملفات الجذر.
- :ext4يمكنك استخدام صورة ext4معدة مسبقًا.
- :fileلنسخ ملفات فردية (النواة، initramfs، البرامج الثابتة .(firmware

في حلوان لينكس:

- SquashFS– .rootfs
- fileللنواة و .initramfs
- في المستقبل، قد نضيف صورة ext4إذا قمنا بتوزيع إصدارات OEM معدة مسبقًا.

3.7.5سير العمل

- يبدأ Calamares وحدة .unpackfs
- يقرأ قائمة العناصر في .unpack
- يستخدم rsyncأو أدوات mountلفك ضغط الصور.
- ينسخ الملفات إلى .rootMountPoint
- يسجل النتائج في .var/log/calamares.log

3.7.6الاستثناءات والتحسينات

باستخدام excludeأو excludeFile، يمكننا:

- تجنب نسخ الملفات غير الضرورية (أدوات التصحيح، المستندات الكبيرة.(
- تقليل وقت التثبيت.
- توفير مساحة على النظام المستهدف.

مثال عملي:

```
exclude:
 - /usr/share/doc/*
 - /usr/share/locale/*
```

3.7.7دراسة حالة حلوان لينكس

الهدف :تجربة تثبيت سريعة (≤ 10 دقائق.(

التصميم:

- **airootfs.sfs:يحتوي على النظام الأساسي.**
- **kernel + initramfs:يتم نسخها مباشرة.**
- **exclude:إزالة اللغات المحلية غير المستخدمة لتقليل الحجم.**

النتيجة :يحصل المستخدم الجديد على نظام جاهز بالكامل مع بصمة تخزين دنيا. وتظل التوزيعة مرنة (يمكن إضافة أو إزالة الحزم داخل ملف ISO نفسه.(

3.7.8المشاكل الشائعة والحلول

- **المشكلة :النظام لا يعمل.**
  - **السبب :غالبًا ما يكون بسبب فقدان النواة.initramfs/**
  - **الحل :تأكد من أن ملفات boot/تم نسخها بشكل صحيح.**
- **المشكلة :التثبيت بطيء.**
  - **السبب :غالبًا ما يكون بسبب نسخ ملفات غير ضرورية.**
  - **الحل :استخدم .exclude**
- **المشكلة :أخطاء .rsync**
  - **الحل :تحقق من المسارات داخل ملف ISO الحي.**

3.7.9أفضل الممارسات

- **اجعل الأمر بسيطًا :ملف rootfsواحد + النواة.**
- **لا تنسخ ملفات كبيرة وغير ضرورية (السجلات، ذاكرة التخزين المؤقت.(**
- **استخدم weightلتوفير تجربة مستخدم أفضل (شريط تقدم واقعي.(**
- **اختبر التثبيت عدة مرات على أجهزة مختلفة للتحقق من السرعة والاستقرار.**

3.7.10خاتمة

تُعد وحدة unpackfsهي المرحلة الأساسية التي تحول حلوان لينكس إلى نظام وظيفي على قرص المستخدم.

بفضل المرونة في تحديد المصادر والوجهات، يمكن لفريق التطوير أن:

- **يبني تجربة تثبيت محسنة.**

- يقلل الوقت والمساحة المستهلكة.
- يضمن أن النظام الناتج مستقر واحترافي.

لذلك، فإن unpackfs.confليس مجرد إعداد في الكواليس؛ إنه العمود الفقري لعملية تثبيت ناجحة.

3.8تشغيل أوامر ما قبل التثبيت باستخدام shellprocess-before.conf

3.8.1مقدمة

أحيانًا، تتطلب عملية تثبيت النظام تنفيذ أوامر محددة قبل نسخ الملفات أو إعداد الإعدادات.

قد تتضمن هذه الأوامر:

- تهيئة مفاتيح الحزم.(pacman-key)
- تحميل مكونات معينة.
- تنفيذ سكربتات خاصة بالتوزيعة.

هنا يأتي دور shellprocess-before.conf، الذي يسمح لك بتعريف قائمة من الأوامر التي سيتم تنفيذها إما داخل النظام الحي (host)أو داخل النظام المستهدف.(target root)

3.8.2الهيكل العام

الملف مكتوب بصيغة YAML ويحتوي على المفاتيح التالية:

- dontChroot:
  - true:يتم تنفيذ الأوامر داخل البيئة الحية (خارج النظام المستهدف.(
  - false:يتم تنفيذ الأوامر داخل النظام المستهدف (الجذر المُثبّت .(/
- timeout:
  - القيمة بالثواني.
  - تحدد المدة المسموح بها لكل أمر أو كقيمة افتراضية لجميع الأوامر.
- script:
  - سلسلة نصية واحدة (أمر واحد.(
  - قائمة من الأوامر (سلاسل نصية أو كائنات.(
  - يمكن أن يبدأ الأمر ببادئة -لتجاهل حالات الفشل.

3.8.3المتغيرات

تدعم الأوامر استبدال المتغيرات:

- **:{ROOT}$المسار إلى النظام المستهدف.**
- **:{USER}$اسم المستخدم المحدد أثناء التثبيت.**

مثال:

```
script:
  - "echo ${USER} >> ${ROOT}/etc/helwan-users"
```

3.8.4مثال من حلوان لينكس

```
YAML
dontChroot: false
timeout: 999
script:
  - "-/usr/bin/pacman-key --init"
  - "-/usr/bin/pacman-key --populate"
```

تحليل:

- **:dontChroot: falseسيتم تنفيذ الأوامر داخل النظام المثبت.**
- **:timeout: 999يتيح وقتًا كافيًا لتجنب حالات الفشل بسبب الأوامر البطيئة.**
- **:pacman-keyيقوم بتهيئة وإعداد مفاتيح Pacman تلقائيًا، مما يضمن أن النظام الجديد يمكنه التواصل مع مستودعات آرتش/حلوان.**
- **البادئة :-إذا فشل أمر (مثل عدم توفر الخادم)، فسيستمر التثبيت بدلاً من التوقف.**

3.8.5سير العمل

- **يقرأ Calamares ملف .shellprocess-before.conf**
- **يحدد ما إذا كان سيعمل في النظام الحي أم داخل النظام المثبت.**
- **ينفذ الأوامر بالترتيب باستخدام .c- sh/bin/**
- **يتم تطبيق مهلة زمنية على كل أمر.**
- **أي خطأ بدون البادئة -سيوقف التثبيت.**
- **يتم تسجيل النتائج في .log.var/log/calamares/**

3.8.6حالات الاستخدام في حلوان لينكس

- **تهيئة مفاتيح pacmanقبل تثبيت الحزم.**

- تشغيل سكربت شل (shell script) يقوم بتعديل الإعدادات الأولية.
- استدعاء خدمة مثل .systemd-machine-id-setup
- إضافة hookمعين أو ملف تهيئة للشبكة.

3.8.7المشاكل الشائعة والحلول

- فشل :pacman-key
  - أضف -لتجاهل حالات الفشل المؤقتة.
- المهلة الزمنية قصيرة جدًا:
  - قم بزيادة القيمة إلى 600 أو 999.
- تشغيل الأوامر في البيئة الخاطئة:
  - تحقق من قيمة) dontChrootمثلاً pacman-key :يجب أن يعمل داخل النظام المستهدف.(

3.8.8أفضل الممارسات

- تجنب إضافة الكثير من الأوامر؛ يجب أن يظل التثبيت سريعًا.
- استخدم -للأوامر غير الحرجة.
- وثّق دائمًا أي تغييرات في التوثيق (للمطورين والمستخدمين.(
- راقب سجل Calamares بحثًا عن أي أخطاء متكررة.

3.8.9خاتمة

قد يبدو ملف shellprocess-before.confبسيطًا، ولكنه يوفر قوة هائلة في التحكم بما يحدث قبل أن يبدأ التثبيت الفعلي.

في حلوان لينكس، يعكس استخدامه لتهيئة pacman-keyفلسفة التوزيعة:

- أتمتة كاملة.
- مرونة عالية.
- أمان واحترافية.

3.9تشغيل عمليات التنظيف بعد التثبيت باستخدام shellprocess-final.conf

3.9.1مقدمة

بعد نسخ الملفات(unpackfs) ، وإعداد المستخدمين(users.conf) ، وتشغيل أوامر ما قبل التثبيت-(shellprocess before.conf)، تأتي خطوة حاسمة:

- **تنظيف النظام الجديد :إزالة الملفات المؤقتة أو الإعدادات التي لم تعد ضرورية بعد عملية التثبيت.**

تتمثل وظيفة shellprocess-final.confفي ضمان أن النظام النهائي الذي يتم تسليمه للمستخدم نظيف، آمن، وخالٍ من أي بقايا لبيئة التثبيت أو السكربتات المؤقتة.

3.9.2الهيكل العام

على غرار الملفات السابقة، هذا الملف مكتوب بصيغة YAML ويحتوي على المفاتيح الأساسية:

- **dontChroot:**
  - **true:ينفّذ الأوامر داخل البيئة الحية.(host)**
  - **false:ينفّذ الأوامر داخل النظام المثبت (مطلوب عادةً.(**
- **timeout:**
  - **يحدد الوقت الأقصى (بالثواني) المسموح به لكل أمر.**
- **script:**
  - **سلسلة نصية واحدة (أمر واحد.(**
  - **قائمة من الأوامر (سلاسل نصية أو كائنات.(**
  - **الأوامر التي تبدأ ببادئة -تشير إلى أن فشلها لن يوقف التثبيت.**
- **المتغيرات المتاحة:**
  - **${ROOT}:المسار إلى المجلد الجذر للنظام المثبت.**
  - **${USER}:اسم المستخدم الذي تم اختياره أثناء التثبيت.**

3.9.3مثال من حلوان لينكس

```
dontChroot: false

timeout: 999

script:

 - "-rm ${ROOT}/etc/sudoers.d/g_wheel"

 - "-rm -r ${ROOT}/etc/mkinitcpio.conf.d/"

 - "-rm -r ${ROOT}/etc/systemd/system/getty@tty1.service.d"

 - "-rm -r ${ROOT}/etc/systemd/system/multi-user.target.wants/pacman-init.service"

 - "-rm -r ${ROOT}/etc/systemd/system/pacman-init.service"

 - "-rm ${ROOT}/etc/systemd/system/etc-pacman.d-gnupg.mount"
```

- "-rm ${ROOT}/root/.automated_script.sh"

- "-rm ${ROOT}/root/.zlogin"

- "-rm ${ROOT}/etc/polkit-1/rules.d/49-nopasswd_global.rules"

- "-rm ${ROOT}/etc/polkit-1/rules.d/49-nopasswd-calamares.rules"

تحليل:

- **pacman-init services: يتم إزالتها لأنها مطلوبة فقط أثناء عملية التثبيت.**
- **sudoers.d/g_wheel: يُستخدم لمنع أي إعدادات sudo غير مرغوبة.**
- **polkit rules (nopasswd): يزيل أي صلاحيات غير آمنة قد تظل في النظام المثبت.**
- **root/.automated_script.sh & .zlogin: ملفات مؤقتة خاصة بنظام ISO الحي.**

النتيجة: نظام نظيف، جاهز للاستخدام، خالٍ من أي ملفات أو إعدادات قديمة.

3.9.4 سير العمل

- **يقرأ Calamares ملف shellprocess-final.conf.**
- **ينفّذ الأوامر واحدًا تلو الآخر داخل النظام المثبت.(dontChroot: false)**
- **أي أمر مسبوق بـ - يتم تجاهل فشله.**
- **بعد التنفيذ، تُسجل جميع العمليات في /var/log/calamares.log.**

3.9.5 حالات الاستخدام في حلوان لينكس

- **إزالة قواعد polkit التي تسمح بـ sudo بدون كلمة مرور.**
- **تنظيف systemd من خدمات التمهيد الخاصة بالبيئة الحية.**
- **حذف ملفات سكربتات التثبيت من حساب الجذر.**
- **ضمان أن المستخدم يتلقى نظامًا نظيفًا، "كما لو كان جديدًا."**

3.9.6 المشاكل الشائعة والحلول

- **بقايا ملفات لم يتم حذفها: تأكد من أن مسارات ${ROOT} صحيحة.**
- **التثبيت يتوقف بسبب أمر فاشل: أضف - قبل الأمر لتجاهل الخطأ.**
- **المهلة الزمنية غير كافية: قم بزيادة قيمة timeout.**

بالتأكيد، إليك الترجمة الكاملة والدقيقة للنص الذي أرسلته.

3.9.7 أفضل الممارسات

- **نفّذ أوامر التنظيف فقط داخل النظام المثبت.**
- **استخدم - للأوامر غير الحرجة.**
- **راجع سجل التثبيت دائمًا للتأكد من نجاح عملية التنظيف.**
- **قلل من عدد الأوامر لتجنب إطالة وقت التثبيت.**

3.9.8 خاتمة

يعمل ملف shellprocess-final.conf كخط دفاع أخير في عملية التثبيت. من خلاله، يمكن لحلوان لينكس أن يضمن أن النظام الذي تم تسليمه للمستخدم هو:

- **آمن.**
- **نظيف من أي بقايا.**
- **جاهز للاستخدام الفوري. وهذا يعكس فلسفة التوزيعة في تقديم تجربة كاملة واحترافية من الخطوة الأولى إلى الأخيرة.**

3.10 تهيئة مديري العرض باستخدام displaymanager.conf

3.10.1 مقدمة

بمجرد تثبيت النظام الأساسي وتنظيفه، تأتي الخطوة الحاسمة التالية: إعداد مدير العرض (Display Manager - DM). مدير العرض هو المسؤول عن توفير شاشة تسجيل الدخول وتهيئة الجلسة الرسومية للمستخدم. يسمح ملف displaymanager.conf في حلوان لينكس بالتهيئة الدقيقة لمديري العرض المتاحين، بالإضافة إلى أي سكربتات إعداد يجب تشغيلها تلقائيًا أثناء التثبيت.

3.10.2 الهيكل العام

ملف الإعدادات مكتوب بصيغة YAML ولديه ثلاثة مفاتيح رئيسية:

- **displaymanagers: قائمة بمديري العرض ليتم تثبيتها أو إتاحتها للاختيار. الأمثلة تشمل sddm و lightdm و gdm.**
- **basicSetup: قيمة منطقية (true أو false) تحدد ما إذا كان يجب تشغيل سكربتات الإعداد الأساسية لمديري العرض تلقائيًا.**
- **sysconfigSetup: قيمة منطقية (true أو false) تشير إلى ما إذا كان يجب تطبيق إعدادات إضافية على مستوى النظام (مثل إعدادات X11 أو Wayland) تلقائيًا.**

3.10.3 مثال من حلوان لينكس

:displaymanagers

sddm -

lightdm -

gdm -

false :basicSetup

false :sysconfigSetup

تحليل:

- **displaymanagers: يمتلك المستخدمون خيارات متعددة متاحة؛ يمكن للمثبت أن يطلب من المستخدم اختيار واحد أو تثبيت كل ما هو مدرج.**
- **basicSetup: false: يمنع تشغيل سكربتات الإعداد الآلية، مما يترك تحكمًا أدق للمستخدمين المتقدمين أو لسكربتات ما بعد التثبيت.**
- **sysconfigSetup: false: يتم تخطي إعدادات مستوى النظام، مما يمنح المستخدم المرونة لتهيئة البيئات الرسومية يدويًا.**

3.10.4 سير العمل

- **أثناء التثبيت، يقرأ Calamares ملف displaymanager.conf.**
- **يتحقق من قائمة displaymanagers ويثبت أو يفعّل مديري العرض المدرجين.**
- **إذا كانت basicSetup أو sysconfigSetup true، فسيتم تنفيذ السكربتات المقابلة.**
- **يسجل سجل التثبيت (/var/log/calamares.log) جميع العمليات للمراجعة.**

3.10.5 حالات الاستخدام في حلوان لينكس

- **السماح للمستخدمين باختيار مدير تسجيل الدخول الرسومي المفضل لديهم.**
- **تخطي الإعدادات الآلية للمستخدمين المتقدمين الذين يريدون تحكمًا كاملاً.**
- **الحفاظ على المرونة لكل من البيئات الرسومية خفيفة الوزن والغنية بالميزات.**

3.10.6 المشاكل الشائعة والحلول

المشكلة	الحل
مدير العرض لا يعمل بعد التثبيت	تحقق من تثبيت مدير العرض وتفعيله عبر.systemctl enable <dm>.service
عدة مديري عرض تسبب تعارضات	أدرج مدير عرض واحد فقط في القائمة أو عطّل مديري العرض الإضافيين يدويًا.
الجلسة الرسومية تفشل	تأكد من تطبيق sysconfigSetup إذا كان مطلوبًا بواسطة مدير العرض المختار.

3.10.7 أفضل الممارسات

- **أدرج فقط مديري العرض التي تنوي دعمها؛ المدراء الإضافيون يمكن أن يخلقوا تعارضات.**

- استخدم basicSetup و sysconfigSetup بحكمة بناءً على المستخدم المستهدف: false للمستخدمين المتقدمين، true للإعدادات الآلية.
- تحقق دائمًا من وظائف مدير العرض بعد التثبيت عن طريق فحص السجلات واختبار تسجيل الدخول.

3.10.8 خاتمة

يضمن ملف displaymanager.conf أن حلوان لينكس يقدم تجربة تسجيل دخول رسومية مرنة واحترافية. من خلال السماح بالتحكم الآلي واليدوي، فإنه يحقق توازنًا بين قابلية الاستخدام والتحكم المتقدم - مما يعكس فلسفة نظام مصقول وجاهز للاستخدام.

3.11 تهيئة Initramfs باستخدام initcpio.conf

3.11.1 مقدمة

بعد تثبيت وتهيئة النظام الأساسي، ومدير العرض، وخطوات التنظيف، تأتي المرحلة الهامة التالية: إعداد initramfs. initramfs هو نظام ملفات جذر مؤقت يتم تحميله في الذاكرة أثناء الإقلاع. وهو يهيئ النظام لتركيب نظام ملفات الجذر الحقيقي ويؤدي مهام الإقلاع المبكرة الأساسية. في حلوان لينكس، يسمح ملف initcpio.conf بتهيئة كيفية قيام mkinitcpio بإنشاء initramfs، بما في ذلك أي النواة التي سيتم بناؤها وما إذا كان يجب فرض تخفيفات أمنية.

3.11.2 الهيكل العام

ملف initcpio.conf هو ملف إعدادات يعتمد على YAML وله مفتاحان رئيسيان في هذا الإصدار:

- kernel:
  - يحدد أي نواة (نوى) يجب على mkinitcpio إنشاء صور initramfs لها.
  - القيم الممكنة:
    - اسم إعداد مسبق واحد، على سبيل المثال linux.
    - فارغ أو غير مضبوط (يبني لجميع النوى).
    - السلسلة النصية الحرفية "all" (يبني لجميع النوى).
- be_unsafe:
  - قيمة منطقية (true أو false) تحدد ما إذا كان سيتم تعطيل التخفيفات (mitigations) الخاصة بصلاحيات الملفات المتساهلة.
  - false (افتراضي): صلاحيات آمنة لملفات initramfs، مما يحمي البيانات الحساسة مثل مفاتيح LUKS.
  - true: يوقف التخفيفات الأمنية، مما قد يكون خطيرًا إذا تم تخزين initramfs على نظام ملفات غير POSIX (مثل قسم EFI).

3.11.3 مثال من حلوان لينكس

linux :kernel

false :be_unsafe

تحليل:

- **kernel: linux: يولد initramfs فقط لإعداد النواة المسبق المسمى linux. وهذا فعال إذا تم تثبيت عدة أنوية ولكن يتم استخدام واحدة فقط.**
- **be_unsafe: false: يضمن بقاء صلاحيات الملفات في initramfs صارمة، مما يحمي المعلومات الحساسة مثل مفاتيح التشفير. وهذا أمر بالغ الأهمية للأمان، خاصة على الأنظمة التي تستخدم LUKS.**

3.11.4 سير العمل

- **أثناء التثبيت أو تحديثات النواة، يقرأ حلوان لينكس ملف initcpio.conf.**
- **يتم استدعاء mkinitcpio وفقًا لمفتاح kernel.**
- **يتم تطبيق صلاحيات الملفات وفقًا لإعداد be_unsafe.**
- **يتم تخزين صورة initramfs الناتجة في /boot/initramfs-<kernel>.img.**
- **يتم تسجيل جميع الإجراءات لاستكشاف الأخطاء وإصلاحها.**

3.11.5 حالات الاستخدام في حلوان لينكس

- **بناء initramfs لنواة واحدة أو عدة أنوية بكفاءة.**
- **ضمان أمان صارم للأقراص المشفرة عن طريق الحفاظ على be_unsafe: false.**
- **تخصيص البناء للمستخدمين المتقدمين الذين قد يعطلون التخفيفات عمدًا.**

3.11.6 المشاكل الشائعة والحلول

المشكلة	الحل
لم يتم توليد Initramfs لنواة معينة	تحقق من أن مفتاح kernel يطابق الإعداد المسبق المطلوب أو اضبطه على."all"
الأقراص المشفرة بـ LUKS تفشل في الفتح	تأكد من ضبط be_unsafe على.false
تخزين initramfs على نظام ملفات غير POSIX	لا يمكن تطبيق التخفيفات الأمنية؛ تأكد من أن القسم آمن أو تقبل المخاطر مع be_unsafe: true.

3.11.7 أفضل الممارسات

- **استخدم kernel لتقييد توليد initramfs على النوى التي تستخدمها بالفعل.**
- **حافظ دائمًا على be_unsafe: false ما لم يكن لديك سبب محدد لتعطيل التخفيفات الأمنية.**
- **بعد تحرير initcpio.conf، أعد توليد initramfs باستخدام mkinitcpio -P.**
- **سجل وتحقق من initramfs لضمان موثوقية الإقلاع.**

3.11.8 خاتمة

يمنح ملف initcpio.conf في حلوان لينكس تحكمًا دقيقًا في توليد initramfs والأمان. من خلال تهيئة هدف النواة بعناية وفرض صلاحيات ملفات آمنة، تضمن التوزيعة سلوك إقلاع مبكر آمن، وموثوق، وقابل للصيانة، مما يعكس فلسفتها الاحترافية وجاهزيتها للمستخدم.

3.12 إزالة مستخدم باستخدام removeuser.conf

3.12.1 مقدمة يُعد ملف removeuser.conf أداة حاسمة في حلوان لينكس لضمان نظافة النظام وأمانه بعد التثبيت. فهو يحدد تلقائيًا ويزيل حسابات المستخدمين المؤقتة أو الافتراضية من النظام النهائي، وهو أمر مفيد بشكل خاص في بيئات OEM أو الجلسات الحية. وهذا يمنع الحسابات المتبقية من تشكيل مخاطر أمنية أو تتعلق بالخصوصية على المستخدم.

3.12.2 الهيكل العام ملف removeuser.conf هو إعداد YAML مباشر مع مفتاح واحد:

- **username: اسم حساب المستخدم المراد حذفه. في حالة الجلسة الحية، يكون هذا عادةً liveuser. تعمل هذه الوحدة باستخدام أمر لينكس القياسي userdel. وهي مصممة لتكون غير حرجة، مما يعني أن أي فشل في حذف المستخدم لن يوقف عملية التثبيت. بدلاً من ذلك، يتم تسجيل هذه الأخطاء ببساطة في /var/log/calamares.log، مما يسمح للتثبيت بالاستمرار دون عوائق.**

3.12.3 مثال من حلوان لينكس

liveuser :username

تحليل: هذا الإعداد البسيط يوجه مثبت Calamares لحذف حساب liveuser بعد تثبيت النظام. وهذا يضمن أن المستخدم النهائي يتلقى نظامًا نظيفًا بدون أي حسابات افتراضية من البيئة الحية.

3.12.4 سير العمل

- **قراءة الإعدادات: خلال مرحلة ما بعد التثبيت، يقرأ Calamares ملف removeuser.conf.**
- **تنفيذ userdel: تحاول الوحدة حذف المستخدم المحدد بواسطة مفتاح username.**
- **تسجيل النتائج: يتم تسجيل أي نجاح أو فشل في ملف سجل التثبيت في /var/log/calamares.log.**
- **مواصلة التثبيت: تستمر عملية التثبيت بغض النظر عن نتيجة حذف المستخدم.**

3.12.5 حالات الاستخدام في حلوان لينكس

- **تنظيف الجلسة الحية: إزالة liveuser المؤقت بعد التثبيت يوفر للمستخدم نظامًا نظيفًا وآمنًا من البداية.**
- **إعدادات OEM أو المخصصة: يسمح هذا بحذف الحسابات المؤقتة أو الافتراضية التي تم إنشاؤها أثناء عملية بناء آلية أو مخصصة.**
- **الأمان: منع الوصول العرضي إلى حسابات المستخدمين القديمة وغير المراقبة.**

3.12.6 المشاكل الشائعة والحلول

- **المشكلة: لا يمكن حذف المستخدم.**
  - **الحل: تحقق مما إذا كان المستخدم مسجلاً دخوله أو لديه عمليات قيد التشغيل. على الرغم من أن Calamares مصمم للتعامل مع هذا gracefully، يمكن استخدام أمر pkill -u <username> اليدوي لإنهاء العمليات قبل الحذف اليدوي.**
- **المشكلة: التثبيت يتوقف بشكل غير متوقع.**
  - **الحل: تم تصميم الوحدة خصيصًا لتجاهل الأخطاء وعدم إيقاف التثبيت. إذا حدث توقف، فمن المحتمل أن يكون ذلك بسبب مشكلة أخرى داخل Calamares، وليس بسبب وحدة removeuser.conf نفسها.**

3.12.7 أفضل الممارسات

- **استهداف الحسابات الضرورية فقط: حدد فقط الحسابات المؤقتة أو غير الضرورية للإزالة. لا تحاول إزالة مستخدمين حرجين للنظام مثل root أو nobody.**
- **مراجعة السجلات: راجع دائمًا سجلات التثبيت للتأكد من تنفيذ خطوة إزالة المستخدم كما هو متوقع.**
- **الجمع مع عمليات التنظيف الأخرى: ادمج removeuser.conf مع وحدات تنظيف أخرى مثل shellprocess-final.conf لضمان أن النظام النهائي مُجهز بالكامل وخالٍ من البيانات غير الضرورية.**

3.12.8 خاتمة يُعد ملف removeuser.conf أداة بسيطة ولكنها حيوية في حلوان لينكس للحفاظ على نظافة النظام وأمانه. من خلال الإزالة التلقائية للحسابات المؤقتة أو الافتراضية، يوفر حلوان لينكس نظامًا جاهزًا للاستخدام وخاليًا من المستخدمين غير الضروريين، مما يعكس فلسفة التوزيعة في الاحترافية والاهتمام بالتفاصيل من التثبيت إلى الاستخدام الأول.

3.13 خاتمة: استكشاف وحدات Calamares

يظل Calamares واحدًا من أهم وأكثر أدوات التثبيت استخدامًا في بيئة لينكس، وتثق به العديد من التوزيعات لمرونته، وموثوقيته، وهيكله النمطي. في هذا الفصل، استكشفنا بعض الوحدات الرئيسية - من shellprocess-final.conf لتنظيف النظام، إلى displaymanager.conf لإعداد تسجيل الدخول الرسومي، و initcpio.conf لتهيئة initramfs، و removeuser.conf لإدارة المستخدمين. ومع ذلك، يوفر Calamares العديد من الوحدات الأخرى التي تتجاوز ما قمنا بتغطيته. توفر كل وحدة إمكانيات إضافية، وخيارات أتمتة، وفرصًا للتخصيص. نشجع القراء على استكشاف النطاق الكامل لوحدات Calamares للاستفادة الكاملة من قوته، وتكييف تجربة التثبيت مع احتياجات توزيعتهم أو مشروعهم. من خلال فهم هذه الوحدات، يوضح حلوان لينكس كيف يمكن تقديم نظام احترافي وجاهز للاستخدام - نظيف، وآمن، ومرن من الإقلاع الأول.

الملحق الأول: أوامر آرتش لينكس الأساسية

يقدم هذا الملحق مرجعًا سريعًا لأوامر آرتش لينكس الأساسية، والتي تُعد حاسمة لأي مستخدم جديد يتنقل في بيئة آرتش. هذه الأوامر هي الأدوات الأساسية للتفاعل مع النظام.

1 . إدارة النظام باستخدام pacman

pacman هو مدير حزم آرتش لينكس القوي.

تحديث النظام بأكمله

Bash

sudo pacman -Syu

- **S: مزامنة (synchronize) الحزم.**
- **y: تحديث قاعدة بيانات الحزم من المستودعات.**
- **u: ترقية جميع الحزم المثبتة إلى أحدث إصداراتها. نصيحة: يوصى بتشغيل هذا الأمر أسبوعيًا للحفاظ على نظامك آمنًا ومحدثًا.**

تثبيت حزمة جديدة

Bash

sudo pacman -S firefox

ينزل هذا الأمر متصفح Firefox ويثبته من مستودعات آرتش لينكس الرسمية.

إزالة حزمة

Bash

sudo pacman -R firefox

يزيل هذا الأمر حزمة firefox ولكنه يترك ملفات إعداداتها سليمة.

إزالة حزمة مع التبعيات وملفات الإعدادات

Bash

sudo pacman -Rns firefox

- **n: يزيل ملفات الإعدادات المرتبطة بالحزمة.**
- **s: يزيل التبعيات غير الضرورية التي لم تعد مطلوبة من قبل أي حزمة أخرى مثبتة. ⚠ تحذير: استخدم العلامة -Rns بحذر. يمكن أن تزيل مكتبات قد تكون ضرورية لبرامج أخرى، مما قد يؤدي إلى تعطيل وظائفها.**

البحث عن الحزم

Bash

```
pacman -Ss vlc
```

يبحث هذا الأمر في قواعد بيانات الحزم ويعرض جميع النتائج المتعلقة بـ VLC (مثل vlc, vlc-plugin-alsa) الموجودة في المستودعات.

عرض تفاصيل الحزمة

Bash

```
pacman -Si vlc
```

يعرض هذا الأمر معلومات مفصلة حول الحزمة، بما في ذلك إصدارها، حجمها، المستودع، التبعيات، الوصف، والمزيد.

تنظيف ذاكرة التخزين المؤقت للحزم

Bash

```
sudo pacman -Sc
```

يزيل هذا الأمر ملفات الحزم القديمة المخزنة مؤقتًا والتي لم تعد مثبتة، مما يساعد على تحرير مساحة على القرص.

2 . إدارة المستخدمين

إضافة مستخدم جديد

Bash

```
sudo useradd -m -G wheel ahmed
```

- **m-: ينشئ مجلد home للمستخدم الجديد (/home/ahmed).**
- **G wheel-: يضيف المستخدم ahmed إلى مجموعة wheel. يُمنح المستخدمون في هذه المجموعة عادةً صلاحيات sudo، مما يسمح لهم بتنفيذ الأوامر كـ superuser.**

تعيين كلمة مرور

Bash

```
sudo passwd ahmed
```

يطالبك هذا الأمر بإدخال وتأكيد كلمة مرور للمستخدم المحدد (ahmed).

منح صلاحيات Sudo لمنح وصول sudo للمستخدمين في مجموعة wheel:

- **افتح ملف sudoers للتحرير:**

Bash

sudo visudo

- **ألغِ التعليق من السطر التالي عن طريق إزالة # في البداية:**
- **# %wheel ALL=(ALL) ALL**

هذا يسمح لأعضاء مجموعة wheel بتشغيل أي أمر كأي مستخدم (بما في ذلك root) باستخدام sudo.

3 . إدارة الملفات والمجلدات

عرض الملفات مع التفاصيل

Bash

ls -l

يعرض هذا الأمر محتويات المجلد الحالي، مع إظهار معلومات مفصلة مثل صلاحيات الملف، المالك، المجموعة، الحجم، وتاريخ التعديل.

نسخ ملف

Bash

/cp file.txt /home/ahmed

ينسخ هذا الأمر file.txt من المجلد الحالي إلى مجلد /home/ahmed/.

نقل أو إعادة تسمية ملف

Bash

mv file.txt file2.txt

يعيد هذا الأمر تسمية file.txt إلى file2.txt في المجلد الحالي. يمكن استخدامه أيضًا لنقل الملفات إلى مجلدات مختلفة.

حذف ملف أو مجلد

Bash

/rm -rf folder

- **r-: يحذف المجلدات ومحتوياتها بشكل متكرر (recursively).**
- **f-: يفرض الحذف دون المطالبة بالتأكيد. ⚠ تحذير بالغ: لا تستخدم أبدًا rm -rf /. سيحذف هذا الأمر نظام ملفات الجذر بأكمله، مما يجعل نظامك غير قابل للإقلاع. استخدمه بحذر شديد.**

⚙4 . إدارة الخدمات باستخدام systemctl

يُستخدم systemctl للتحكم في خدمات النظام (daemons).

بدء/إيقاف خدمة

Bash

sudo systemctl start NetworkManager

sudo systemctl stop NetworkManager

تبدأ هذه الأوامر وتوقف خدمة NetworkManager، على التوالي.

تفعيل/تعطيل خدمة عند الإقلاع

Bash

NetworkManager enable sudo systemctl

NetworkManager disable sudo systemctl

- **enable: يهيئ الخدمة للبدء تلقائيًا عند إقلاع النظام.**
- **disable: يمنع الخدمة من البدء تلقائيًا عند الإقلاع.**

التحقق من حالة الخدمة

Bash

systemctl status NetworkManager

يعرض هذا الأمر الحالة الحالية للخدمة (مثلاً: نشطة، غير نشطة، فاشلة)، بالإضافة إلى سجلات الدخول الأخيرة.

5 . إدارة الشبكة

عرض أجهزة الشبكة المتاحة

Bash

ip link

يعرض هذا الأمر جميع واجهات الشبكة المتاحة على النظام (مثل eth0 لـ Ethernet، wlan0 لـ Wi-Fi).

تفعيل بطاقة شبكة لاسلكية

Bash

wlan0 up set sudo ip link

ينشط هذا الأمر واجهة الشبكة اللاسلكية wlan0، مما يجعلها جاهزة للاستخدام.

التحقق من الاتصال

Bash

ping archlinux.org

يرسل هذا الأمر طلبات صدى ICMP إلى المضيف المحدد (archlinux.org) لاختبار اتصال الشبكة ووقت الاستجابة (lat

ency). اضغط على Ctrl+C للإيقاف.

ملخص الملحق الأول غطى هذا الملحق:

- **أوامر pacman لتحديثات النظام، وتثبيت الحزم، وإزالتها، والبحث، وتنظيف ذاكرة التخزين المؤقت.**
- **أساسيات إدارة المستخدمين والصلاحيات، بما في ذلك إضافة المستخدمين، وتعيين كلمات المرور، ومنح صلاحيات sudo.**
- **عمليات الملفات والمجلدات الأساسية مثل العرض، والنسخ، والنقل، والحذف.**
- **التحكم في الخدمات باستخدام systemctl لإدارة daemons النظام.**
- **أوامر إدارة الشبكة الأساسية للتحقق من الأجهزة والاتصال.**

نصيحة: هذا الملحق بمثابة مرجع سريع لأي مستخدم جديد لآرتش لينكس. اعتبره "مجموعة أدواتك الأولى" قبل التعمق في عالم حلوان لينكس.

الملحق الثاني: اختصارات أوامر حلوان لينكس

يوفر حلوان لينكس مجموعة من الاختصارات المدمجة (aliases) لتبسيط إدارة النظام والتعامل مع الحزم. تلتف هذه الاختصارات حول أوامر pacmanالخاصة بآرتش لينكس، مما يسمح للمستخدمين بأداء المهام المتكررة بسرعة أكبر وبشكل أكثر سهولة .بدلاً من كتابة الأوامر الطويلة، يمكن للمستخدمين الاعتماد على هذه الأشكال المبسطة لتوفير الوقت وتقليل الأخطاء.

1. اختصارات إدارة الحزم

مزامنة قواعد بيانات الحزم

sync

المكافئ sudo pacman -Syyy :يفرض تحديثًا كاملاً لجميع قواعد بيانات الحزم من المستودعات.

تثبيت حزمة

install firefox

المكافئ sudo pacman -S firefox :يثبت حزمة مباشرة من المستودعات.

تحديث النظام

update

المكافئ sudo pacman -Syyu :يحدث قواعد البيانات ويرقي جميع الحزم المثبتة.

البحث عن حزم في المستودعات

search vlc

المكافئ sudo pacman -Ss vlc :يبحث في المستودعات البعيدة عن الحزم.

البحث عن الحزم المثبتة محليًا

search-local firefox

المكافئ sudo pacman -Qs firefox :يعرض الحزم المثبتة محليًا التي تطابق الاستعلام.

عرض معلومات الحزمة

pkg-info vlc

المكافئ sudo pacman -Qi vlc :يعرض معلومات مفصلة عن حزمة مثبتة.

تثبيت حزمة محلية

local-install ./package.zst

المكافئ sudo pacman -U ./package.zst :يثبت ملف حزمة من نظام الملفات المحلي (على سبيل المثال، حزمة .(zst.

مسح ذاكرة التخزين المؤقت للحزم

clr-cache

المكافئ sudo pacman -Scc :يزيل الحزم المخزنة مؤقتًا لتحرير المساحة.

فتح قاعدة بيانات pacman

unlock

المكافئ sudo rm /var/lib/pacman/db.lck :يزيل ملف القفل إذا تم مقاطعة ⚠ .pacmanاستخدم فقط إذا كنت متأكدًا من أن pacmanلا يعمل.

إزالة حزمة

remove firefox

المكافئ sudo pacman -R firefox :يزيل الحزمة المحددة مع ترك ملفات الإعدادات الخاصة بها.

إزالة حزمة مع الإعدادات والتبعيات

autoremove firefox

المكافئ sudo pacman -Rns firefox :يزيل الحزمة مع ملفات الإعدادات والتبعيات غير المستخدمة ⚠ .استخدم بحذر لتجنب تعطيل برامج أخرى.

2. معلومات النظام

عرض معلومات النظام

helwan

المكافئ uname -a :يطبع معلومات مفصلة عن النظام مثل إصدار النواة، البنية، واسم المضيف.

ملخص الملحق الثاني

قدم هذا الملحق اختصارات أوامر حلوان لينكس، والتي صممت من أجل:

- **تبسيط استخدام pacman.**
- **توفير الوقت وجهد الكتابة.**
- **توفير تجربة سهلة الاستخدام مع الحفاظ على قوة آرتش لينكس الكاملة .معًا مع الملحق الأول (أساسيات آرتش لينكس)، تشكل هذه الاختصارات مجموعة أدوات عملية للمستخدمين الجدد والمتقدمين في حلوان لينكس على حد سواء.**

الملحق الثالث: مرجع أمر hpm

مدير حزم حلوان (hpm) هو أداة مساعدة سهلة الاستخدام حول pacmanو مساعدات .AUR يسمح للمستخدمين بإدارة البرامج بلغات متعددة، مما يجعل التجربة سهلة الوصول لجمهور عالمي .كل أمر يدعم:

- **اختصار :اسم مستعار من حرف واحد للاستخدام السريع.**
- **الإنجليزية :اللغة الافتراضية.**
- **العربية :واجهة عربية كاملة.**
- **الإسبانية.Soporte en español :**
- **الصينية.中文支持 :**

1. □الأوامر الأساسية

الأمر (الصينية(	الأمر (الإسبانية(	الأمر (العربية(	اختصار	الأمر (الإنجليزية(
安装	instalar	تثبيت	i	install
卸载	eliminar	إزالة	r	remove
升级	actualizar	ترقية	u	upgrade
刷新	sincronizar	تحديث	s	refresh
搜索	buscar	بحث	q	search
信息	informacion	معلومات	I	info
列表	lista	قائمة	l	list
清理	limpiar	تنظيف	c	clean
孤儿	huerfanos	يتيم	o	orphans
AUR	aur	aur	a	aur
诊断	diagnostico	فحص	d	doctor
历史	historial	سجل	h	history
--帮助	--ayuda	--مساعدة	-	--help

2. أمثلة الاستخدام

تثبيت حزمة

hpm install firefox

hpm i firefox

hpm تثبيت firefox

hpm instalar firefox

hpm 安装 firefox

إزالة حزمة

hpm remove vlc

hpm r vlc

hpm إزالة vlc

hpm eliminar vlc

hpm 卸载 vlc

البحث عن حزمة

hpm search gimp

hpm q gimp

hpm بحث gimp

hpm buscar gimp

hpm 搜索 gimp

ملاحظات

- **الاختصارات (i, r, u, إلخ) تعمل بجميع اللغات للاستخدام السريع.**
- **الدعم متعدد اللغات يضمن أن حلوان لينكس متاح عالميًا.**
- **يمكن للمستخدمين العرب التفاعل مع النظام بلغتهم الأم، بينما يتمتع المجتمع الإسباني والصيني أيضًا بدعم كامل.**

ملخص الملحق الثالث

تجسد أداة hpm فلسفة حلوان لينكس:

- **بساطة من خلال الاختصارات.**
- **إمكانية الوصول مع الدعم متعدد اللغات.**
- **قوة من خلال دمج pacman وAURفي واجهة واحدة .مع hpm، تصبح إدارة الحزم في حلوان لينكس سلسة - سواء كنت تفكر بالإنجليزية، عربي، Español، أو中文.**

الملحق الرابع: إصلاحات وحلول سريعة (أسئلة شائعة(

في هذا الملحق، جمعنا أكثر المشاكل شيوعًا مع آرتش/حلوان لينكس وحلولها العملية. الهدف هو أن يجد القارئ المشكلة والحل في بضعة أسطر فقط، دون بحث طويل.

1. ⚠ النظام لا يعمل بعد التثبيت

- **السبب المحتمل :مشكلة في تثبيت GRUB أو خطأ في تقسيم القرص.**
- **الحل:**
  - **قم بالتمهيد إلى البيئة الحية من ملف.ISO**
  - **غيّر الجذر إلى نظامك المثبت.arch-chroot /mnt :**
  - **أعد تثبيت-GRUB: grub-install --target=x86_64-efi --efi-directory=/boot --bootloader id=GRUB.**
  - **أعد إنشاء ملف إعدادات.GRUB: grub-mkconfig -o /boot/grub/grub.cfg**

2. Wi-Fi ❌ لا يعمل

- **السبب المحتمل NetworkManager :غير مفعل أو أن تعريف بطاقة Wi-Fi مفقود.**
- **الحل:**
  - **تحقق من بطاقة الشبكة.ip link :**
  - **فعّل .NetworkManager: sudo systemctl enable --now NetworkManager**
  - **إذا استمرت المشكلة، قم بتثبيت التعريف المناسب من) AUR على سبيل المثال، .(rtl8821ce**

3. ❌ التثبيت بطيء جدًا (مرايا بطيئة(

- **السبب المحتمل :مرايا الحزم بعيدة جغرافيًا أو بطيئة.**
- **الحل :استخدم Reflector لاختيار أسرع المرايا save-- sort rate-- latest 5 --"country "EG --sudo reflector : /etc/pacman.d/mirrorlist.**

4. ❌ الصوت لا يعمل

- **السبب المحتمل PulseAudio :أو PipeWireغير مثبت أو غير مفعل.**
- **الحل:**
  - **ثبّت PipeWire ومكوناته pipewire-pulse pipewire-alsa pipewire -S pacman sudo : pavucontrol.**
  - **أعد تشغيل النظام.**

- افتح pavucontrolوقم بتعيين جهاز الإخراج الافتراضي الخاص بك.

5. شاشة سوداء بعد التثبيت

- **السبب المحتمل :**مشكلة في تعريف بطاقة الرسومياتNVIDIA) ، في معظم الحالات.(
- **الحل:**
  - حاول الدخول إلى) TTY وحدة تحكم نصية.Ctrl + Alt + F2 :(
  - ثبّت التعريف المناسب.sudo pacman -S nvidia nvidia-utils :
  - أعد التشغيل.

6. نسيت كلمة مرور المستخدم

- **الحل:**
  - قم بالتمهيد إلى البيئة الحية.
  - نفّذ.arch-chroot /mnt passwd username :
  - قم بتعيين كلمة مرور جديدة.

7. خطأ "فشل في تنفيذ المعاملة"

- **السبب المحتمل :**تعارضات في الحزم أو ترقية جزئية.
- **الحل:**
  - قم بتحديث النظام بالكامل.sudo pacman -Syu :
  - إذا استمرت المشكلة، حاول إزالة الحزمة المتعارضة، ثم أعد تثبيتها.

8. الفأرة أو لوحة المفاتيح لا تعمل

- **السبب المحتمل :**تعريف مفقود.
- **الحل:**
  - تأكد من تثبيت الحزمة التالية.sudo pacman -S xf86-input-libinput :

9. مساحة القرص ممتلئة فجأة

- **السبب المحتمل :**ذاكرة تخزين pacmanمؤقتة كبيرة.
- **الحل.**sudo paccache -r :
  - يحذف هذا الأمر جميع إصدارات الحزم القديمة، مع ترك الإصدارات الثلاثة الأحدث فقط.

ملخص الملحق الرابع

- **يمكن لأي مستخدم يواجه مشكلة شائعة أن يجد حلاً هنا من خلال خطوات سريعة قليلة.**
- **إنه يوفر وقتًا كبيرًا عن طريق تجنب البحث المكثف في Arch Wiki أو المنتديات.**
- **إنه بمثابة "إسعافات أولية" لأي حالة طوارئ في حلوان لينكس.**

الملحق الخامس: موارد إضافية للتعلم والدعم

يعتمد حلوان لينكس على آرتش، لذا فإن أقوى مصدر للتعلم هو آرتش نفسه، بالإضافة إلى مواردنا الخاصة. فيما يلي قائمة منسقة بالمواقع، والقنوات، والكتب، والأدوات التي يمكن أن تساعد أي مبتدئ أو مطور.

1. المواقع الرسمية

- **Arch Wiki: https://wiki.archlinux.org**
  - **المرجع الأكثر شمولاً ودقة لأي شيء يتعلق بآرتش لينكس. إذا كانت لديك مشكلة، فمن المرجح أن يكون الحل هنا في 90% من الأوقات.**
- **صفحة حلوان لينكس الرسمية(https://helwan-linux.github.io/helwanlinux/index.html) :**
  - **توفر تحديثات، وشروحات محددة، وملفات.ISO**
- **منتديات آرتش لينكسhttps://bbs.archlinux.org :**
  - **مكان رائع للمناقشات واستكشاف الأخطاء وإصلاحها. يشتهر مجتمع آرتش بكونه قويًا جدًا.**

2. قنوات يوتيوب

- **SMA Coding (S.M.A):**
  - **تقدم شروحات متعمقة عن لينكس، وحلوان لينكس، والبرمجة بشكل عام.**
- **DistroTube:**
  - **قناة تغطي توزيعات مختلفة وإعدادات لينكس المتقدمة.**
- **Learn Linux TV:**
  - **تقدم دروسًا عملية واحترافية عن لينكس وآرتش.**

3. كتب مفيدة

- **The Arch Linux Handbook:من تأليف مجتمع آرتش.**
- **How Linux Works:يشرح البنية الداخلية للنظام.**
- **Linux Command Line and Shell Scripting Bible:ضروري لفهم Bash بشكل احترافي.**

4. أدوات تطوير التوزيعات

- **GitHub / GitLab:لمتابعة تطوير المشروع.**
- **QEMU / VirtualBox:لاختبار التوزيعة.**
- **Reflector:لتحسين المرايا في آرتش.**

- **Calamares Installer Docs: لفهم وتخصيص Calamares.**

5. مجتمع حلوان لينكس

- **مستودع GitHub: موقع الكود والأدوات – hpm) مدير حزم حلوان.)**

6. نصائح للتعلم الذاتي

- **ابدأ صغيرًا :جرب الأوامر الأساسية كل يوم بدلاً من الشعور بالإرهاق بكم هائل من المعلومات.**
- **وثّق بنفسك :كل مشكلة أو حل تجده - اكتبه في دفتر ملاحظات أو ملف نصي.**
- **شارك مع الآخرين :أفضل طريقة لترسيخ المعلومات هي شرحها لشخص آخر.**
- **جرب واكسر الأشياء :صُمم آرتش وحلوان لتتعلم من أخطائك.**

ملخص الملحق الخامس

- **هذا الملحق هو بوابة لمزيد من التعلم بعد الكتاب.**
- **يمكن لأي قارئ الاعتماد عليه لتطوير نفسه من مستخدم إلى مطور توزيعة.**
- **إنه يكمل الكتاب بدلاً من إنهائه، ويعمل كـ "جسر" إلى المرحلة التالية.**

الملحق السادس: مسرد المصطلحات التقنية

أنظمة التشغيل والتوزيعات

- **لينكس :(Linux) نظام تشغيل مجاني ومفتوح المصدر يعتمد على نواة لينكس.**
- **توزيعة :(Distro) إصدار جاهز للاستخدام من لينكس يتضمن النواة، والحزم الأساسية، وأدوات التثبيت.**
- **آرتش لينكس :(Arch Linux) توزيعة شهيرة تعتمد على مبادئ البساطة والمرونة.**
- **حلوان لينكس :(Helwan Linux) توزيعة مبنية على آرتش، لكنها مهيأة مسبقًا لتكون أسهل للمستخدمين.**

إدارة الحزم

- **:Pacmanمدير الحزم الرسمي لآرتش لينكس.**
- **مستودع :(Repo) موقع مركزي يحتوي على الحزم والبرامج.**
- **:AUR (Arch User Repository)مستودع مجتمعي يحتوي على حزم غير رسمية.**
- **:hpm (Helwan Package Manager)أداة ذكية لتبسيط العمل مع Pacman باستخدام لغة بسيطة وصديقة للإنسان.**

التثبيت والبناء

- **:Archisoأداة لبناء صور ISO مخصصة لتوزيعات آرتش.**
- **:ISOملف صورة نظام يمكن نسخه على USB أو تشغيله في.VirtualBox / QEMU**
- **:Calamaresأداة تثبيت رسومية سهلة الاستخدام تدعم التخصيص.**
- **:Chrootأداة لتغيير المجلد الجذر للنظام مؤقتًا والدخول إلى النظام المثبت لإصلاحه.**

النظام والعمليات

- **النواة :(Kernel) قلب نظام لينكس، المسؤول عن التفاعل مع الأجهزة والموارد.**
- **نظام التمهيد) :(Init System) مثل (systemdالبرنامج المسؤول عن بدء الخدمات عند الإقلاع.**
- **خدمة :(Service) برنامج يعمل في الخلفية (مثل .(NetworkManager**
- **محمل الإقلاع :(Bootloader) برنامج يسمح للنظام بالإقلاع (مثل.(GRUB**

بيئات سطح المكتب ومديرو النوافذ

- **بيئة سطح المكتب :(Desktop Environment - DE) بيئة رسومية كاملة مثلGNOME ، Cinnamon، أو.XFCE**
- **مدير النوافذ :(Window Manager - WM) مدير نوافذ بسيط بدون مكونات إضافية، مثل i3wm أو.Fluxbox**
- **:Cinnamonبيئة سطح مكتب مستقرة وسهلة الاستخدام.**
- **:i3wmمدير نوافذ متقدم يعتمد على لوحة المفاتيح.**

- **Fluxbox: مدير نوافذ خفيف جدًا ومناسب للأجهزة القديمة.**

الأمان

- **Sudo: أداة تسمح بتنفيذ الأوامر بصلاحيات المدير (root).**
- **Root: المستخدم الأساسي ذو الصلاحيات الكاملة في النظام.**
- **جدار الحماية (Firewall): أداة لحماية النظام من الاختراق (مثل UFW).**

الملفات والأنظمة

- **نظام الملفات (File System): طريقة لتخزين وإدارة الملفات (مثل ext4، btrfs).**
- **تركيب (Mount): عملية جعل قسم أو مجلد متاحًا للنظام.**
- **قسم (Partition): تقسيم القرص الصلب إلى أقسام منفصلة للتثبيت أو التخزين.**

التطوير والبرمجة

- **الصدفة (Shell): واجهة سطر أوامر مثل Bash أو Zsh.**
- **سكربت (Script): ملف نصي يحتوي على أوامر ليتم تنفيذها تلقائيًا.**
- **Git: نظام للتحكم في الإصدارات لتتبع التغييرات في الكود.**
- **Makepkg: أداة لإنشاء حزم آرتش من ملفات PKGBUILD.**

ملخص الملحق السادس

- **هذا المسرد هو أداة مرجعية سريعة.**
- **لا يتعين على القارئ حفظ كل شيء؛ يمكنه العودة إليه إذا واجه مصطلحًا جديدًا.**
- **إنه مهم جدًا للمبتدئين ويجعل قراءة بقية الكتاب أسهل.**

ملاحظة :هذا الكتاب لا يزال نسخة تجريبية وهو عرضة للتحسين، وإعادة التنظيم قبل الإصدار النهائي.