

مكتبات ريفية تطوّر برمجياتها: كيف نشأ نظام (Libki) المفتوح المصدر؟

Open Source Library Software Development in a Small Rural Library System

كايل هول، سيندي مردوك إيمز، جون برايس



الملخص:

يعرض المقال تجربة شبكة مكتبات ريفية صغيرة في مقاطعة كروفورد بينسلفانيا في تبني البرمجيات المفتوحة المصدر وتطويرها. ويتبع الانتقال إلى لينكس ونظام (Koha)، ثم تطوير (Libki) لإدارة استخدام الحواسيب العامة، مع بيان التكاليف والوفورات ومتطلبات التدريب والتوظيف. ويوضح كيف يمكن الاستثمار في المهارات الداخلية والتطوير التدريجي لبناء حلول تلائم احتياجات المكتبات وتفيد مجتمعها الأوسع.

الكلمات المفتاحية: البرمجيات المفتوحة المصدر، المكتبات الريفية، إدارة أجهزة الحاسوب العامة، نظام (Koha)، الحد من التكاليف

نستعرض تجربة شبكة مكتبات مقاطعة كروفورد الاتحادية (Crawford County Federated Library System) في تطوير (Libki)، وهو نظام مفتوح المصدر لإدارة استخدام محطات الحاسوب العامة. ومن خلالها نوضح تحديد المشروع ومواصفاته وكتابه برمجياته واختبارها وإنتاجها، ثم متابعة تطويرها. ويتحقق (Libki) من هوية المستخدم ويتيح له استخدام الإنترنت ضمن مدة محددة. ويعمل تطبيق المستخدم على أنظمة متعددة، منها ويندوز ولينكس، بينما تتيح واجهة الإدارة مراجعة السجلات وضبط الوقت والصلاحيات وإنهاء الجلسات عند الحاجة. وقد طوّر موظفو الشبكة النظام بالكامل، وتستخدمه مكتبات في أنحاء الولايات المتحدة.

المقدمة:

كيف تطور شبكة مكتبات في مقاطعة ريفية بينسلفانيا برمجيات مفتوحة المصدر تستفيد منها مكتبات أخرى؟ ولماذا تتولى شبكة كروفورد، التي تضم تسع مكتبات وتخدم أقل بقليل من مئة ألف نسمة، هذا العمل أصلاً؟ يشرح المقال الإجابة تفصيلاً، أما خلاصة موقفنا فهي أننا نستطيع ذلك، ونرى أن المكتبات الأخرى تستطيع أيضاً.

تبنت المكتبات الأمريكية الحواسيب ما إن أصبحت التقنية صالحة للاستخدام عملياً. ولسوء الحظ، كانت الحواسيب في ذلك الوقت ضخمة، وتتطلب مهندسين من معهد ماساتشوستس للتقنية (MIT) لكتابة برمجياتها، وكانت باهظة التكلفة. وكان من الأفضل ترك الدراية اللازمة لإعداد هذه النظم وإدارتها ودعمها للشركات الكبرى، على أن تشتري المكتبات خدماتها. فحتى أكبر المكتبات في البلاد لم تكن قادرة على تطوير نظم الإعارة الخاصة بها. وحتى حين كوّنت المكتبات اتحادات لتطوير البرمجيات، كان تسويق نظم الإعارة وتوزيعها مكلفاً جداً. ولكل هذه الأسباب، آلت نظم الإعارة، التي أصبحت في النهاية النظم المتكاملة للمكتبات (Integrated Library Systems, ILS)، إلى الجهات التجارية في نهاية المطاف.

أما اليوم فقد أصبحت التقنية في كل مكان؛ ولغات البرمجة في متناول الأفراد الحاصلين على التدريب المناسب، والبرمجة تُدرّس في كل جامعة ذات حجم معقول. وتستطيع مجتمعات المكتبات المتقاربة في التوجه أن تتعاون عبر الإنترنت في جميع أنحاء العالم مجاناً. وما إن يُطوّر مشروع مفتوح المصدر ويُطلق حتى يمكن توزيعه، مجاناً أيضاً، ويمكن تحديثه وتغييره مع إسهام مزيد من المكتبات بآرائها. وفي نهاية المطاف تنشأ حول المشروعات المفتوحة المصدر الناجحة مجتمعات نابضة بالحياة. وقد أفضت كل هذه الاتجاهات إلى امتلاك المكتبات اليوم الأدوات التي تحتاج إليها لتطوير برمجياتها الخاصة وصيانتها وتوزيعها، والاستفادة من عمل الآخرين أيضاً. فعلى سبيل المثال، تستفيد أي مكتبة تستخدم خادم الويب أباتشي (Apache) لاستضافة موقعها من منهجية المصدر المفتوح.

ونقترح، أثناء قراءة تجربتنا، التفكير في العوائق التي تحول دون تبني مؤسستك نهجاً مماثلاً. ونرجح أن كثيراً منها يرتبط بأساليب العمل الموروثة، أكثر من ارتباطه بحدود ما تتيحه التقنية الحالية والعمل مفتوح المصدر.

الطريق إلى (Libki):

تستخدم شبكة مكتبات مقاطعة كروفورد الاتحادية البرمجيات المفتوحة المصدر (Open Source Software, OSS) في عملياتها لتقنية المعلومات منذ نحو عام 1999، بدءاً ببعض الموجّهات البسيطة القائمة على الأقراص المرنة، التي استُخدمت لتقاسم اتصال إنترنت هاتفي (dialup) بين حواسيب كل مكتبة. (يُرجى الاطلاع على رابط المصدر للحصول على قائمة كاملة بجميع مشروعاتنا في البرمجيات المفتوحة المصدر.) وبعد نجاح ذلك المشروع أدرّكنا إمكانات البرمجيات المفتوحة المصدر، فدمجناها في بنيتنا التحتية بقدر متزايد.

ومنذ ذلك المشروع الأول قررنا أن يعتمد أي حل جديد لتقنية المعلومات نقدّمه لمكتباتنا الأعضاء مستقبلاً على البرمجيات المفتوحة المصدر كلما أمكن ذلك. واتخذنا هذا القرار لعدد من الأسباب. أولها الموثوقية. فقد كنا محبطين من المنتجات التجارية التي لا نستطيع الاطلاع على طريقة عملها الداخلية، نعتمد فيها على إعادة التشغيل، أو إعادة التثبيت من جديد، أو الدعم الفني الهاتفي لمحاولة حل المشكلات، وكثيراً ما كان ذلك دون جدوى. ووجدنا في النهج العملي المباشر لحل المشكلات في البرمجيات المفتوحة المصدر بديلاً عملياً؛ فبقليل من البحث على الويب أو بطرح الأسئلة في القوائم البريدية أو قنوات (IRC) المناسبة، كنا نجد عموماً حلولاً لمعظم المشكلات التي واجهناها. وثانيها أن تطوير حلولنا الخاصة بالبرمجيات المفتوحة المصدر أتاح لنا التحكم في مشتريات المعدات. فقد اضطررنا مرات عديدة إلى تخصيص أموال ثمينة لاستبدال حواسيب سليمة تماماً لأنها لم تعد تفي بمواصفات أحدث البرمجيات التجارية. وباختيارنا استخدام البرمجيات المفتوحة المصدر على الحواسيب المكتبية والخوادم، تمكّنّا من إطالة عمر كثير من معدّاتنا. أما المسألة الثالثة فهي التخصيص؛ فقد كان بإمكاننا تخصيص البرمجيات المفتوحة المصدر لتلبية احتياجاتنا، من إحكام قفل الحواسيب العامة إلى إنشاء مواقع وخدمات مخصصة

لمكتباتنا. وأخيرًا، كان السبب الأخير هو التكلفة. فإلى جانب توفير المال بإطالة عمر معدّاتنا من خلال استخدام البرمجيات المفتوحة المصدر، تمكّنّا من توفير مبالغ كبيرة لأنها مجانية في الغالب، خلّاقيًا لمعظم البرمجيات التجارية، ولا تتطلب دفع رسوم ترخيص لكل مستخدم أو لكل جهاز.

بالطبع لا يوجد حل مجاني تمامًا؛ فمع البرمجيات المفتوحة المصدر قد توفر المال في البرمجيات ورسوم الترخيص، لكن ثمة تكاليف مرتبطة بتوظيف موظفين قادرين على تثبيتها وإدارتها. غير أننا وجدنا منذ مشروعنا الأول في البرمجيات المفتوحة المصدر فصاعدًا أن تكاليف هؤلاء الموظفين قد تكون أقل بكثير مما تدفعه مؤسسات كثيرة مقابل الحلول البرمجية التجارية. فضلًا عن ذلك، تمكّن موظفو تقنية المعلومات لدينا من تطبيق المهارات والمعارف التي اكتسبوها في تنفيذ حلول البرمجيات المفتوحة المصدر في مكتباتنا على طيف واسع من المشروعات، لأن مشروعات البرمجيات المفتوحة المصدر كثيرًا ما تُبنى باستخدام الأدوات نفسها.

أنجز مشروعنا الأول في البرمجيات المفتوحة المصدر، وهو الوصول المشترك إلى الإنترنت عبر الاتصال الهاتفي لمكتباتنا، عن اضطرار، لأن المنظومة كانت قد كتبت طلب منحة استنادًا إلى برمجية تبين أنها مجرد وعود لم تتحقق (vaporware). ففي أواخر تسعينيات القرن العشرين، قبل توفر النطاق العريض في المجتمعات الريفية، كنا بحاجة إلى إنشاء موجّهات اتصال هاتفي لمكتباتنا الريفية السبع من أجل توفير الوصول إلى الإنترنت للموظفين والجمهور. وقدّرنا تكلفة استخدام إعداد تجاري من البرمجيات والمعدات بنحو 3,150 دولارًا لكل مكتبة، أو نحو 22,000 دولار إجماليًا، وهو مال لم يكن لدى المنظومة. ومن خلال معارف محليين (إذ كانت إحدى أمينات مكتباتنا الريفية متزوجة من أستاذ جامعي تعلّم لينكس ليبنى محطة أرصاد جوية خاصة به) وجدنا خيرًا في لينكس دون مقابل. وعثر هذا المستشار على مشروع موجّه لينكس (Linux Router Project, LRP)، وهو توزيع لينكس قائمة على الأقراص المرنة يمكنها تحويل حاسوب قديم إلى موجّه وجدار حماية. فاشترينا عدة حواسيب قديمة من طراز 486 عبر موقع إيباي (Ebay)، بنحو 40 دولارًا للحاسوب الواحد. ونزعنا منها الأقراص الصلبة، وركّبنا أجهزة مودم بسرعة (56k)، وهكذا وبكل بساطة أصبح لدينا وصول شبكي مشترك إلى الإنترنت وجدار حماية. وما كان رائدًا في هذا الحل أن النظم كانت محصنة نسبيًا ضد الاختراق (إذ صُلبت الأقراص المرنة للقراءة فقط) وسهلة الصيانة جدًّا. وبلغت التكلفة الإجمالية للمشروع أقل من 1,000 دولار. وأُحيل آخر خادِم الاتصال الهاتفي إلى التقاعد بعد خمس سنوات من الخدمة الخالية من المتاعب.

وإدراكًا منا أننا نستطيع توفير مبالغ كبيرة باستخدام البرمجيات المفتوحة المصدر، وأننا لا نستطيع الحصول على المساعدة المجانية طوال الوقت، تعاقدنا بعد ذلك مع خبير لينكس نفسه لتدريب سيندي مردوك (Cindy Murdock Ames)، مديرة تقنية المعلومات لدينا، على استخدام لينكس. وبلغت التكلفة الإجمالية للتدريب أقل من 2,000 دولار.

وباتباع استراتيجية استخدام مواردنا الداخلية للحلول المفتوحة المصدر، بدأنا في استبدال حلولنا المملوكة. وكان المشروع التالي بعد مشروع الموجّهات القائمة على الأقراص المرنة استبدال خادم الويب (Microsoft IIS) لدينا بخادم قائم على لينكس يعمل بأباتشي. وبعد ذلك بوقت قصير، في يناير 2000، استبدلنا حواسيبنا العامة العاملة بنظام ويندوز بحل للعملاء النحيفين (thin clients) قائم على لينكس يسمى (LTSP) (مشروع خادم المحطات الطرفية للينكس، (Linux Terminal Server Project))، رابط المصدر). وتوفر الأجهزة الطرفية الخفيفة القائمة على (LTSP) للرواد الوصول إلى الإنترنت، إضافة إلى حزمة برامج مكتبية (حاليًا (LibreOffice)). ونستخدمها أيضًا حواسيب للفهرس العام داخل المكتبة.

قد يوحى سرد هذه الخطوات بأنها كانت سهلة، لكنها لم تكن كذلك. فقد استغرق تطوير مشروع (LTSP) وحده أكثر من ستة أشهر، لأنه كان أول مشروعنا الكبيرة، ولأن تعلم أدواته تطلب جهدًا كبيرًا. لكن المهارات التي اكتسبناها انتقلت إلى مشروعات أخرى، منها بناء الموجهات، وترشيح المحتوى على محطات الإنترنت العامة، وتخصيص حواسيب الموظفين والجمهور، وإنشاء المواقع وخوادم الملفات.

أما الوفورات التي حققها مشروع (LTSP)، فقد وقّرنا مبلغًا كبيرًا طوال عمر المشروع. فبدلًا من شراء حواسيب عامة جديدة، اشترينا معدات لترقية حواسيبنا القديمة، مع إعادة استخدام الهياكل ووحدات تزويد الطاقة الموجودة. وبلغت تكلفة ترقية المعدات لأجهزتنا الطرفية الخفيفة السبعة الأصلية إضافة إلى خادم (LTSP) ما مجموعه 3,779.25 دولارًا (إذ كلف الخادم 1,485 دولارًا، وكلف كل جهاز طرفي خفيف مرقى 327.75 دولارًا). وبلغ متوسط تكلفتنا لكل جهاز لخمس عشرة عميلًا نضيفًا إضافة إلى الخادم 426.75 دولارًا؛ في حين بلغ السعر الذي حصلنا عليه للعدد نفسه من الأجهزة الطرفية الخفيفة القائمة على ويندوز مع الخادم 1,649 دولارًا لكل جهاز؛ ووفقًا لموقع (CNET) كان متوسط تكلفة الحاسوب الشخصي العامل بنظام ويندوز في يناير 2000 هو 873 دولارًا. غير أننا لو اخترنا حواسيب شخصية تعمل بنظام ويندوز، لتحملنا إضافة إلى ذلك تكلفة برمجيات لإحكام قفل الحواسيب للاستخدام العام، وهو ما تمكّننا من تحقيقه أصلًا مع لينكس. واستخدمنا تلك الحواسيب المُرَقَّة أكثر من عشر سنوات قبل أن تُستبدل بمعدات أحدث، وإن كنا قد استبدلنا الخادم مرتين خلال تلك السنوات العشر. وقد وفّر عدم الاضطرار إلى شراء حواسيب جديدة كل ثلاث سنوات وبرمجيات جديدة كل ثلاث إلى خمس سنوات ما لا يقل عن 31,000 دولار لمكتبة واحدة فقط في منظومتنا (دورنا استبدال للمعدات وثلاث دورات للبرمجيات والتراخيص لخمس عشرة محطة طرفية) على مدى عشر سنوات. وخلال هذه المدة لم نزد عدد الموظفين، وأنجز عمل التطوير لمشروعات البرمجيات المفتوحة المصدر إلى جانب الروتين المعتاد لصيانة النظم والتدريب.

وتوجّنا التحول إلى البرمجيات المفتوحة المصدر بالانتقال من النظام المتكامل للمكتبات (Winnebago CIRC/CAT) إلى النظام المتكامل للمكتبات (Koha) في عام 2005. وبلغت التكلفة الفعلية للانتقال من (Winnebago) إلى (Koha) ما يزيد قليلًا على 45,000 دولار لتسع مكتبات (مكتبتان متوسطتا الحجم وسبع مكتبات ريفية). وكانت هذه التكلفة متماشية مع ما عرضه علينا مورّدون آخرون للنظم المملوكة مقابل الانتقال. ولن تكون تكلفة الانتقال على الأرجح بهذا الارتفاع اليوم لمكتبة تعمل وحدها بالكامل على تنفيذ (Koha)؛ فمعظم تكلفتنا كانت في التعاقد مع إحدى الشركات الداعمة لنظام (Koha) لدمج نظام الفهرسة (Zebra) من شركة (IndexData) في (Koha)، حتى يتمكن من دعم مكتبات ذات مجموعات بحجم مجموعتنا أو أكبر. وتضمنت تكاليفنا الأولية شراء خادم ووقت البرمجة والتطوير الذي أمضاه الموظفون؛ فقد طوّرنّا واجهة أمامية خاصة بنا للموظفين في (Koha)، إضافة إلى كثير من التقارير المخصصة، إذ كان لدينا وصول كامل إلى بياناتنا مع (Koha). ولم تكن ثمة في ذلك الوقت تكاليف جارية مرتبطة بنظام (Winnebago CIRC/CAT)، لأنه لم يعد مدعومًا من شركة (Spectrum)، الشركة المالكة له حاليًا. وكنا قد اطلعنا على العروض الحالية لشركة (Spectrum)، لكننا رأينا أنه بالنظر إلى نجاحاتنا السابقة مع البرمجيات المفتوحة المصدر، يمكننا أن نجني من الانتقال إلى (Koha) أكثر مما نجنيه من التعامل مع مورّد تجاري مرة أخرى. ولأن (Koha) قائم بالكامل على الويب، فلن نكون مقيدين باستخدام ويندوز على حواسيب الإعارة والفهرس، فتمكّنّا كذلك من تحويل حواسيب موظفينا والفهارس العامة إلى حواسيب مكتبية تعمل بـ لينكس أو أجهزة طرفية خفيفة، فوفّرنا بذلك رسوم تراخيص ويندوز مستقبلاً، وأعطينا الموظفين من دعم بيئة حواسيب مكتبية تعمل بـ ويندوز. فضلًا عن ذلك، لأننا نستطيع دعم (Koha) داخليًا، فلا توجد رسوم دعم سنوية. والتكلفة الجارية الوحيدة مع (Koha) هي ترقية المعدات المخطط لها والموظفون.

وعند الانتقال إلى (Koha)، وظفنا كايل هول ضمن فريق تقنية المعلومات. وأتاحت لنا خبرته ودرجة الماجستير التي يحملها في تقنية المعلومات توجيه التطوير إلى احتياجات الشبكة. وهكذا استخدمنا ما وفرناه من تراخيص البرمجيات للاستثمار في الموظفين ومهاراتهم.

ومن الجوانب المثيرة للاهتمام في استخدام نظام متكامل مفتوح المصدر للمكتبات أننا تمكّنًا من تثبيته نظامًا تجريبيًا لإثبات الفكرة (proof of concept)، بتصدير البيانات من نظامنا المتكامل الحالي إلى تثبيت تجريبي لنظام (Koha). وباستخدام الخوادم التجريبية هذه، تمكّن موظفو مكتباتنا من تجربة (Koha) بالقدر الذي يرغبون فيه دون الحاجة إلى شراء أي برمجيات. كما تمكّنّا من إجراء تجارب متكررة لعملية الانتقال للتأكد من أن جميع بياناتنا نُقلت كما هو متوقع. وتمكّنّا أيضًا من نقل كل مكتبة من مكتباتنا من تثبيتها الفردي لنظام (Winnebago) إلى خادم (Koha) المركزي لدينا وفق جدولنا الزمني الخاص، على مدى عدة أشهر. ولو تعاقدنا مع شركة لنقلنا إلى نظام متكامل مملوك للمكتبات، لربما لم نحظّ بتلك المرونة. فعند نقل كل مكتبة، كنا نوقف نظامها القديم يوم الجمعة أو السبت عند الإغلاق، وننقل جميع بياناتها، ثم تفتح أبوابها يوم الاثنين مستخدمة (Koha) وجميع بياناتها من تسجيلات (MARC) والمستفيدين والمعاملات سليمة.

ومع انضمام مطور إلى الفريق، بحثنا عن أفضل استخدام لوقته. خُصصت المرحلة الأولى لتطوير (Koha) وتحسينه وتكييفه لاحتياجاتنا، ثم اقترح المكتبيون إيجاد طريقة أفضل لتنظيم مدة استخدام الحواسيب العامة.

ولأن المنظومة كانت تستخدم البرمجيات المفتوحة المصدر منذ عدة سنوات على معظم حواسيبها العامة، كان من المنطقي استخدام برمجيات مفتوحة المصدر لإدارة وقت المستفيدين على تلك الحواسيب. فجميع الحواسيب العامة تقريبًا في مكتباتنا التسع إما محطات عمل مستقلة تعمل ب لينكس أو أجهزة طرفية خفيفة قائمة على لينكس. وقد حاولنا في البداية استخدام حلول قائمة على البرمجيات المفتوحة المصدر لإدارة وقت المستفيدين، لكن أيًا منها لم يلبّ احتياجاتنا، فاتخذنا قرارًا بتطويره بأنفسنا.

كيف طُوّر (Libki):

الإلهام:

بدأ (Libki) من رغبة كايل هول في أتمتة التسجيل لاستخدام حواسيب الإنترنت العامة. فقد كانت المكتبات تعتمد ورقة يكتب فيها المستفيد اسمه ووقت بدء الجلسة، ويضطر المكتبي إلى مراجعتها باستمرار، ثم مطالبة من انتهى وقته بترك الجهاز للمستفيد التالي. وكانت هذه المتابعة تضيف عبئًا إلى العمل اليومي.

البحث:

بحثنا عن نظام لإدارة محطات الحاسوب العامة وفق ثلاثة متطلبات أساسية: أن يعمل على ويندوز ولينكس، وأن يكون حرًا ومفتوح المصدر بالكامل، وأن يستخدم رقم الباركود وكلمة المرور المعتمدين في (Koha) للتحقق من الهوية. واشترط المكتبيون كذلك إمكانية إضافة وقت للجلسة، ومعرفة مستخدمي الأجهزة، وإرسال الرسائل إليهم، وإنهاء الجلسات عند الضرورة.

قيّمنا أولًا الحلول المفتوحة المصدر القائمة لنرى ما إذا كانت ستلبي احتياجاتنا.

كان المرشح الأول (OpenKiosk). وبدا أن هذه البرمجية تطابق المتطلبات تمامًا. فقد أُطلقت شيفرتها المصدرية بموجب رخصة جنو العمومية (GNU Public License)، وكان العميل يعمل على لينكس وويندوز كليهما. غير أن مراجعة أعمق كشفت أن المؤلف، وإن كان ربما التزم بنص الرخصة، فقد انتهك روحها. فالنصف الخاص بالعميل من النظام كان يُترجم (compile) على لينكس، لكن ترجمته على ويندوز كانت تتطلب ملفًا خاصًا لم يُتَح. وكان الحصول على نسخة من العميل متوافقة مع ويندوز يتطلب الدفع مقابل كل إصدار. ونقل من موقع المؤلف:

يستخدم عميل ويندوز الشيفرة المصدرية نفسها لعميل يونكس (Unix)، لكنه غير مرتبط بأي مكتبة (Qt) مرخصة بموجب رخصة (GPL). غير أنه موصول بطبقة (Win32) مكتوبة بلغة (C++) وشبيهة بمكتبة (Qt). وبهذه الطريقة يمكن تقاسم الشيفرة المصدرية بين عمليي يونكس وويندوز دون أي تغييرات كبيرة. [1]

وقد أدى هذا، مقترنًا بإعلان المؤلف أنه سيتوقف عن التطوير، إلى التخلي عن (OpenKiosk) بوصفه خيارًا.

وكان المرشح الثاني، (OutKafe)، أكثر وعدًا بكثير. فهو شبيه جدًا في خصائصه بـ (OpenKiosk)، وكان بالفعل برمجية حرة مفتوحة المصدر بالكامل، دون أي شروط خفية. وفي الاختبار بدا أن البرمجية تعمل دون مشكلات. وتضمن تصميم (OutKafe) قاعدة بيانات لأسماء المستخدمين وكلمات مرورهم وحالاتهم، وبرمجية العميل التي تعمل فوق نظام تشغيل (ويندوز أو لينكس)، وبرمجية الإدارة. غير أنه تبين أن برمجية الإدارة، وإن كانت متوافقة مع نظام تشغيل لينكس الذي كنا نستخدمه آنذاك، لم تكن قادرة على العمل على النحو الصحيح على خادم (LTSP) الذي كانت ستُشَر عليه.

البداية: تكييف النظام القائم:

ولما لم تكن ثمة خيارات مفتوحة المصدر قائمة صالحة، اتُخذ القرار بتكييف (OutKafe) لتلبية احتياجاتنا. وبدأ المشروع مجرد بديل قائم على الويب لبرنامج الإدارة في (OutKafe)، مع الإبقاء على بقية تطبيق (OutKafe). وقد صُمم عميل (OutKafe) وخادمه بحيث لا يتفاعلان مباشرة، بل عبر الوصول إلى قاعدة البيانات باستخدام نظام إدارة قواعد البيانات (Postgres). فكان كل من جانبي البرمجية يتصل مباشرة بقاعدة بيانات (OutKafe) ويُجري عليها تحديثات يستطيع الجانب الآخر قراءتها بعد ذلك. ومع أن هذا لم يكن أفضل تصميم، فقد سهّل كتابة واجهة إدارية بديلة. وأُطلقت هذه الواجهة الإدارية الجديدة برمجية حرة مفتوحة المصدر بموجب رخصة (GPL) تحت اسم (outkafe-web). وهكذا تألف نظامنا لإدارة محطات الحاسوب العامة في هذه المرحلة من مكونين (العميل وقاعدة البيانات) مكتوبين ضمن (OutKafe)، ومكوّن واحد (الإدارة) كتبه كايل هول.

وسرعان ما اتضح أن العملية المسؤولة في (OutKafe) عن إنقاص الوقت المستخدم يجب أن تُعاد كتابتها. ثم جاء طلب بتخصيص واجهة العميل. ولم تكن هذه التخصيصات ممكنة مع العميل بصورته المكتوبة، ولم يكن كايل ملماً بلغة البرمجة (باسكال، Pascal) التي كُتِب بها العميل.

ولادة (Libki):

لم يلبّ الحل المعدل احتياجات المكتبيين كاملة، فقررنا تطوير المكونات الثلاثة بأنفسنا: الإدارة وتطبيق المستخدم وإدارة قاعدة البيانات. وبعد نحو تسعة أشهر من التطوير المتقطع، استُبدلت أجزاء (OutKafe) الأصلية كلها. وأطلق كايل البدائل مجتمعة باسم (Libki)، وهو نظام حر مفتوح المصدر لإدارة محطات الحاسوب العامة.

تُبت (Libki) أولًا في مكتبة المقر الرئيس للشبكة، وخضع لاختبارات تجاوزت ثلاثة أشهر لإصلاح العيوب وتدريب الموظفين. وبعد استقرار العمل به، عرضناه على المكتبات الأخرى، فاختار معظمها استخدامه.

ومع أن النظام لم يكن مثاليًا، فقد عمل، وما زال يعمل، لدى المكتبات في شبكة مكتبات مقاطعة كروفورد الاتحادية وفي أماكن أخرى. ومع أن عدد المؤسسات التي تستخدم (Libki) غير معروف، فقد تواصل معنا بعضها ليشكرنا، وقدّم بعضها إسهامات دعمًا لتطوير (Libki).

التطوير اللاحق:

مع أن البرنامج كان صالحًا للعمل بما يكفي لأن تستخدمه مكتبات كثيرة على مر السنين منذ إنطلاقه، أدرك المطور أن في البرمجية عددًا من الجوانب التي يمكن تحسينها.

فقد كان كل من العميل والخادم يتطلب وصولًا مباشرًا إلى قاعدة البيانات، وهو أثر متبقي من كونه بديلًا لبرمجية مختلفة. وكانت واجهة الإدارة على الويب تتطلب إعادة تحميل الصفحة كاملة لتحديث البيانات عن المستخدمين والعملاء. وبعبارة أخرى، إذا تركت صفحة الإدارة مفتوحة أي مدة من الزمن، فسرعان ما تصبح البيانات قديمة. إضافة إلى ذلك، كانت برمجية العميل بطيئة التحميل.

أدت هذه القيود إلى إعادة بناء النظام باسم (Libki 2.0). فأعيد تطوير تطبيق المستخدم بلغة (C++) وباستخدام إطار (Qt4)، لتحسين سرعة التشغيل والتكامل مع نظام التشغيل. كما أُضيفت طبقة توافق (shim) إلى الخادم القائم تتيح للتطبيق التواصل معه مباشرة، بدل الاتصال المباشر بقاعدة البيانات.

يجري حاليًا تطوير نظام إدارة جديد يعمل عبر الويب مثل سابقه، مع تحسينات واسعة في الواجهة والأداء. ويستخدم تقنية (AJAX) لتحديث المعلومات المعروضة باستمرار، ويتيح عرض سجل استخدام الأجهزة، مع نظام صلاحيات يضبط قدرة الموظفين على تغيير الإعدادات.

ومن التحسينات الكبيرة الأخرى في (Libki 2.0) دعم بروتوكول (SIP2). فالإصدار الحالي من (Libki) يتكامل مع النظام المتكامل للمكتبات في المقاطعة بالاستفادة من الطبيعة المفتوحة المصدر لنظام (Koha). فقد تمكّن كاييل من تعديل (Koha) ليرسل تنبيهات إلى خوادم (Libki) في أنحاء المقاطعة من أجل إضافة الحسابات وتحديثها وحذفها كما هي موجودة في نظام (Koha). وبإضافة دعم (SIP2) إلى النظام، يستطيع (Libki) ببساطة أن يرسل طلبًا إلى النظام المتكامل للمكتبات للتحقق من هوية مستخدم معين، ثم يتصرف بناءً على النتائج. ويتيح هذا التطوير أيضًا لنظام (Libki) التكامل بالدخول الموحد مع أي نظام متكامل للمكتبات يدعم (SIP2).

والوضع الحالي لنظام (Libki 2.0)، حتى نوفمبر 2012، أنه يعمل على خادم تجريبي استعدادًا للاختبار التجريبي (Beta) في المقر الرئيس للمنظومة. وسيستغرق الاختبار التجريبي على الأرجح نحو شهر قبل إنطلاقه للاستخدام الفعلي. والشيفرة الحالية لنظام (Libki 2.0) متاحة على موقع (SourceForge.net) عبر (Libki.org). ومن التحسينات المستقبلية الممكنة في هذه المرحلة حجز محطات الحاسوب العامة، إضافة إلى إدارة الطباعة والدفع.

استراتيجية الشبكة وإمكان تطبيقها في مكتبات أخرى:

تجاوز الاستثمار الإجمالي في المشروع خمسة عشر ألف دولار بقليل، شاملاً العمل والمعدات، وكان وقت كابل هول أكبر عناصر التكلفة. وتقع الشبكة في شمال غرب بنسلفانيا، حيث تنخفض تكاليف المعيشة مقارنة بمناطق أمريكية كثيرة؛ ولذلك قاربت تكلفة موظف التطوير ما تدفعه مكتبة حضرية كبيرة لمكتبي في الدرجة المهنية الأولى.

يقوم نهجنا على تحديد الحاجة، وفحص الحلول المفتوحة المتاحة، وتقدير الحاجة إلى التطوير، ثم بناء برنامج ونموذج تجريبي، وجمع ملاحظات المكتبيين. وبعد ذلك نختبر نسخة في إحدى المكتبات، قبل إتاحتها لبقية المكتبات الراضية في استخدامها.

ولا يُرصد لوقت التطوير الفعلي بند في الميزانية، بل هو مجرد واحد من مهام كابل. فقد كان كابل، في وقت مشروع (Libki)، لا يكتب الشيفرة فحسب، بل يثبت أيضاً النظم الحاسوبية ويصونها في جميع مكتبات المقاطعة، بما في ذلك مهام مثل تمديد كابلات الشبكة، وتركيب أنظمة الأمن، والبحث عن برمجيات يمكن تبنيها في مكتباتنا. وكان التطوير يُنجز خارج المهام الروتينية الأخرى أو في الأوقات التي لا تكون فيها زيارة المكتبات الأخرى أمراً عملياً، كما في الطقس السيئ. وهو يعمل حالياً في التطوير لصالح المنظومة بدوام جزئي، بينما يعمل بدوام كامل مطوراً لنظام (Koha) لدى شركة (ByWater Solutions).

قد يبدو أن تجربتنا يصعب تكرارها؛ فقد وظفنا مبرمجاً بأجر يقل عن أسعار السوق، وأسندنا إليه المهام التقنية المعتادة، وأتينا له التطوير حين يسمح وقته. لكن الصورة الأخرى هي أننا وجدنا سبيلاً إلى التطوير رغم قلة الموظفين والتمويل، وصعوبة استقطاب الكفاءات إلى منطقة منخفضة الأجور طويلة الشتاء.

وكانت الاستراتيجية التي اتبعناها هي تدريب موظفينا الحاليين على أنظمة التشغيل المفتوحة المصدر، واستخدام جميع مصادر الدعم المجاني المتاحة، وتوظيف أفراد أذكى من الجامعات الحكومية المحلية ممن يرضون بالبقاء في منطقتنا، وتطوير مشروع واحد في كل مرة دون موعد نهائي ثابت لطرحه. فهل ستج هذه الطريقة بعينها في كل مكتبة في البلاد؟ لا. لكن ما نود قوله هنا في مقاطعة كروفورد هو أن أدوات البرمجيات المفتوحة المصدر والأفراد المدربين على استخدامها أصبحوا متاحين اليوم أكثر من أي وقت مضى. وبالمزج والمواءمة بين أحدث الأدوات والموارد، تستطيع أي مكتبة أن تضع استراتيجية فعّالة لتطوير البرمجيات المفتوحة المصدر أو تبنيها. ولا يتطلب ذلك إلا منظوراً جديداً، واستعداداً للتفكير بحلول تتجاوز الممارسات المعتادة، ودعمًا كاملاً من الإدارة من أجل صياغة خطة فعّالة لتطوير البرمجيات المفتوحة المصدر للمكتبات و/أو تبنيها.

الخاتمة:

إجمالاً، امتد تطوير (Libki) على مدى خمس سنوات، بما يقدر بنحو 500 ساعة من وقت التطوير. ومقابل هذا الجهد، حظيت المنظومة باستخدام برنامج فعّال ومجاني لإدارة محطات الحاسوب العامة. والبرنامج متاح مجاناً لكل من يرغب في استخدامه، ونحن نعلم أن مكتبات عديدة استفادت من استخدامه. ويشاركنا ممثل عن مكتبات مقاطعة كوس (Coos County Libraries) تجربته:

نستخدم (Libki) في خمس مكتبات عامة ومكتبة أكاديمية واحدة في اتحادنا. وقد مكن (Libki) موظفي المكتبات من التخلص من أعباء إدارة الحواسيب العامة المشتتة للانتباه، وأتاح لهم تركيز جهودهم على خدمات الإعارة والخدمات المرجعية. وبفضل واجهته السهلة الاستخدام للموظفين، ودعمه للبرمجيات المفتوحة المصدر وويندوز على السواء، يُعدّ (Libki) حلًا مثاليًا لإدارة وقت الحواسيب العامة في المكتبة العامة.

هل وُفّرت المنظومة المال بهذا المشروع؟ إن حلًا مغلق المصدر لتسع مكتبات على مدى المدة نفسها سيقترّب من المبلغ نفسه الذي أنفقناه على تطوير حلّنا الخاص. غير أن لدينا الآن نظامًا لإدارة محطات الحاسوب العامة يلبي احتياجاتنا تمامًا. فإذا ظهرت مشكلات نستطيع إصلاحها بأنفسنا، فلا انتظار لإتاحة الإصدار التالي من أجل إصلاح الأخطاء البرمجية، ولا اعتماد على خدمات دعم خارجية. يُضاف إلى ذلك أنه مع نضج المنتج وإسهام مكتبات أخرى في تطوير مشروع (Libki)، يتناقص استثمارنا السنوي. فإذا نظرت إلى القيمة الإجمالية لنظام (Libki) بالنسبة إلى المنظومة في السنوات الخمس الأولى، فقد كان مشروعًا متعادل التكلفة تقريبًا؛ لكننا في المستقبل سنوفّر المال. وهذا هو المعتاد في مشروعات مفتوحة المصدر أخرى أنجزناها. فالتكلفة الأولية مماثلة للحلول التجارية؛ لكن الوفورات اللاحقة قد تكون هائلة، ويمكن استثمارها في تطوير مشروعات أخرى، فضلًا عن أن مجتمع المكتبات يجني عائدًا يتمثل في برنامج جديد مفتوح المصدر.

يمكن القول إن مشروع (Libki) يُظهر ما قد يسوء في التطوير المفتوح المصدر. فلم نجد في البداية برنامجًا مفتوح المصدر جاهزًا يلبي متطلبات مشروع بسيط لإدارة الوقت. وبتغيير جانب واحد من برنامج قائم، تمكّنّا من جعله يعمل. وما إن جعلناه يعمل حتى وجدنا أننا نحتاج إلى قدرات إضافية، فاضطررنا إلى مزيد من التطوير إلى حد أننا اضطررنا في النهاية إلى إعادة كتابة بقية البرنامج. وما إن شغلنا ذلك الحل الثاني حتى قررنا أن نجعله أفضل، وأن نضيف إلى البرنامج أحدث التقنيات (Ajax) والمعايير (SIP2). وهكذا، مرة أخرى، أعدنا بناء (Libki) لنجعله أفضل. فللوصول إلى ما وصلنا إليه، اضطررنا في الأساس إلى إعداد ثلاثة إصدارات من البرنامج. فهل كان الأفضل لنا أن نكتب برنامجًا جديدًا من البداية؟ ربما، لكن الأرجح أننا ما كنا لنصيب في المرة الأولى على أي حال. والنتيجة النهائية أن لدى المنظومة الآن برنامجًا حديثًا لإدارة محطات الحاسوب العامة يرتبط بسلسلة بنظامنا المتكامل (Koha)، وكل ذلك بتكلفة نظام تجاري مماثل لم يكن بالضرورة ليلبي جميع متطلباتنا.

والجانب المشرق في كل ما بُذل حتى الآن من عمل ومال هو أن الوصول المجاني إلى (Libki) لم يعد متاحًا للمنظومة وحدها، بل لبقية مجتمع المكتبات أيضًا. وهذا بالضبط نوع التعاون الذي ينبغي لجميع المكتبات أن تنظر فيه. ونحن نعلم أن العذر الذي سيقدّم هو أن ليس كل المكتبات قادرة على تحمّل تكلفة توظيف مطوّر. لكن إذا كان علينا جميعًا أن نوظف محاسبين لقسم الشؤون المالية أو مفرسين للخدمات الفنية، أفلا ينبغي أن نعدّ المبرمجين أيضًا عنصرًا حيويًا في المكتبة؟ أفلا ينبغي أن تُضمّن مهارات المطوّر وقدراته في الهيكل التنظيمي للمكتبة؟ نحن هنا في غابات بنسلفانيا الريفية نؤمن بأن الإجابة عن هذا السؤال هي نعم.

المراجع:

1. <http://openkiosk.sourceforge.net/dw.html>