Software Engineering

Software Evolution & Maintenance

Part 4
(Program comprehension & Reverse Engineering)

T.A Mohammed Sultan

Modified from Sommerville's originals

Software Engineering

هندسة البرمجيات

Software Evolution & Maintenance Part 4)Program comprehension & Reverse Engineering (

T.A Mohammed Sultan

تطور البرمجيات وصيانتها الجزء الرابع)فهم البرنامج والهندسة العكسية (ط. محمد سلطان

Modified from Sommerville’s originals

معدلة من أصول سومرفيل

Program comprehension

Program comprehension

- concerned with studying the way software engineers understand programs.

Objective of studying program comprehension

- design tools that will facilitate the understanding of large programs.

Program comprehension Program comprehension

فهم البرنامج فهم البرنامج

 concerned with studying the way software engineers understand programs.

 يهتم بدراسة طريقة فهم مهندسي البرمجيات للبرامج.

Objective of studying program comprehension

الهدف من دراسة فهم البرنامج

 design tools that will facilitate the understanding of large programs.

 تصميم الأدوات التي من شأنها تسهيل فهم البرامج الكبيرة.

Program Comprehension Strategies

- The bottom-up model
- The top-down model
- The integrated model

Program Comprehension Strategies

استراتيجيات فهم البرنامج

- Thebottom-upmodel
- النموذج السفلي
- Thetop-downmodel
- النموذج العلوي والسفلي
- The integrated model
- النموذج المتكامل

The bottom-up model

- Comprehension starts with the source code and abstracting from it to reach the overall comprehension of the system.
- Steps:

  - ❑ *Read the source code*
  - ❑ *Mentally group together low-level programming details (chunks) to build higher-level abstractions*
  - ❑ *Repeat until a high-level understanding of the program is formed*

The bottom-up model

النموذج التصاعدي

Comprehension starts with thesource code and abstracting fromittoreachthe overallcomprehensionofthesystem.

يبدأ الفهم بالكود المصدري والتجريد منه للوصول إلى الفهم الشامل للنظام.

Steps:

الخطوات:

- Read the source code

- اقرأ الكود المصدري

 Mentally group together low-level programming details (chunks) to build higher-level abstractions

 قم بتجميع تفاصيل البرمجة ذات المستوى المنخفض (أجزاء) عقليًا معًا لبناء تجريدات ذات مستوى أعلى

 Repeat until a high-level understanding of the program is formed

 كرر ذلك حتى يتم تكوين فهم عالي المستوى للبرنامج

The top down model

- Comprehension starts with a general idea, or hypothesis, about how the system works
- Often obtained from a very quick look at what components exist

- Steps
  - *First formulate hypotheses about the system functionality*
  - *Verify whether these hypotheses are valid or not*
  - *Create other hypotheses, forming a hierarchy of hypotheses*
  - *Continue until the low-level hypotheses are matched to the source code and proven to be valid or not*

The top down model

النموذج من أعلى إلى أسفل

Comprehension starts withageneral idea,orhypothesis,about how the system works

يبدأ الفهم بفكرة عامة أو فرضية حول كيفية عمل النظام

Often obtained from avery quick look at what components exist

غالبًا ما يتم الحصول عليها من خلال نظرة سريعة جدًا على المكونات الموجودة

Steps

خطوات

- First formulate hypotheses about the system functionality
- قم أولاً بصياغة فرضيات حول وظيفة النظام
- Verify whether these hypotheses are valid or not
- التحقق من صحة هذه الفرضيات أم لا
- Create other hypotheses, forming a hierarchy of hypotheses
- إنشاء فرضيات أخرى، وتشكيل تسلسل هرمي للفرضيات
- Continue until the low-level hypotheses are matched to the source code and proven to be valid or not
- استمر حتى تتم مطابقة الفرضيات ذات المستوى المنخفض مع الكود المصدري وإثبات صحتها أو عدم صحتها

The Integrated Model

- Combines the top down and bottom up approaches.

- Empirical results show that maintainers tend to switch among the different comprehension strategies depending on

  - ❑ *The code under investigation*
  - ❑ *Their expertise with the system*

The Integrated Model

النموذج المتكامل

• Combines thetop down and bottom up approaches.

• يجمع بين النهج من أعلى إلى أسفل ومن أسفل إلى أعلى.

• Empirical resultsshow that maintainers tendtoswitch among the different comprehension strategies depending on

• تظهر النتائج التجريبية أن المشرفين يميلون إلى التبديل بين استراتيجيات الفهم المختلفة اعتماداً على ذلك

 The code under investigation

 الكود قيد التحقيق

 Their expertise with the system

 خبرتهم بالنظام

Partial program comprehension

- Usually is not necessarily to understand the whole system if only part of it needs to be maintained. But a high fraction of bugs arise from not understanding enough!

Most software maintenance tasks can be met by answering **seven** basic questions:

- ❑ *How does control flow reach a particular location?*
- ❑ *Where is a particular subroutine or procedure invoked?*
- ❑ *What are the arguments and results of a function?*
- ❑ *Where is a particular variable set, used or queried?*
- ❑ *Where is a particular variable declared?*
- ❑ *What are the input and output of a particular module?*
- ❑ *Where are data objects accessed?*

Partial program comprehension

الفهم الجزئي للبرنامج

Usuallyisnot necessarily tounderstand thewhole system if only part of it needs to be maintained.But a high fraction of bugs arise from not understanding enough!

ليس من الضروري عادة فهم النظام بأكمله إذا كان هناك حاجة إلى صيانة جزء منه فقط. ولكن نسبة كبيرة من الأخطاء تنشأ من عدم الفهم الكافي!

Mostsoftware maintenance tasks canbemet by answering seven basicquestions: How does control flow reach a particular location?

يمكن تحقيق معظم مهام صيانة البرامج من خلال الإجابة على سبعة أسئلة أساسية: كيف يصل تدفق التحكم إلى موقع معين؟

 Where is a particular subroutine or procedure invoked?

 أين يتم استدعاء روتين فرعي أو إجراء معين؟

 What are the arguments and results of a function?

 ما هي وسائط ونتائج الدالة؟

 Where is a particular variable set, used or queried? Where is a particular variable declared?

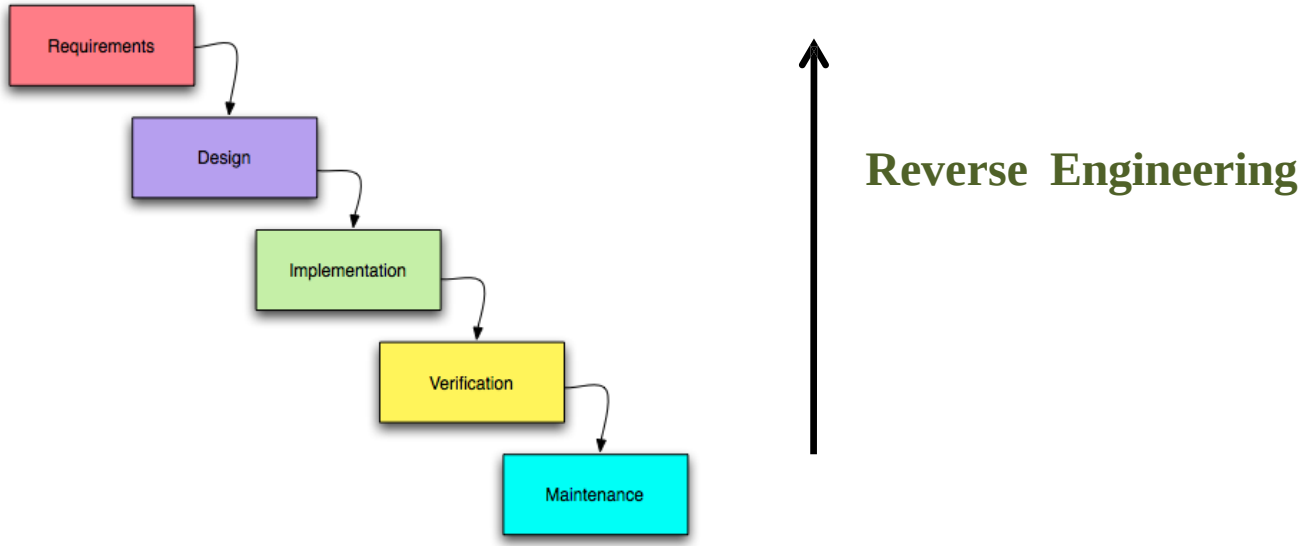
 أين يتم تعيين أو استخدام أو الاستعلام عن متغير معين؟ أين يتم الإعلان عن متغير معين؟

 What are the input and output of a particular module? Where are data objects accessed?

 ما هي المدخلات والمخرجات لوحدة معينة؟ أين يتم الوصول إلى كائنات البيانات؟

Reverse Engineering

- *Process of analyzing a subject system to create representations of the system at a higher level of abstraction.*
- *Going backwards through the development cycle.*



Reverse Engineering

الهندسة العكسية

Process of analyzing a subject system to create representations of the system at a higher level of abstraction.

عملية تحليل نظام موضوعي لإنشاء تمثيلات للنظام على مستوى أعلى من التجريد.

Going backwards through the development cycle.

العودة إلى الوراء من خلال دورة التطوير.

Two main levels of reverse engineering

Binary reverse engineering

- *Take a binary executable*
  - *Recover source code you can then modify*
- *Useful for companies that have lost their source code*
- *Used extensively by hackers*
- *Can be used legally, e.g. to enable your system to interface to existing system*
- *Illegal in some contexts*

Source code reverse engineering

- *Take source code*
  - *Recover high level design information*
- *By far the most widely performed type of reverse engineering*

Two main levels of reverse engineering

مستويين رئيسيين للهندسة العكسية

Binary reverse engineering

الهندسة العكسية الثنائية

• Take a binary executable

• خذ الملف الثنائي القابل للتنفيذ

• Recover source code you can then modify

• استرداد كود المصدر ويمكنك بعد ذلك تعديله

• Useful for companies that have lost their source code

• مفيد للشركات التي فقدت كود المصدر الخاص بها

• Used extensively by hackers

• يستخدم على نطاق واسع من قبل المتسللين

• Can be used legally, e.g.

• يمكن استخدامها بشكل قانوني، على سبيل المثال.

to enable your system to interface to existing system

لتمكين نظامك من التفاعل مع النظام الحالي

• Illegal in some contexts

• غير قانوني في بعض السياقات

Source code reverse engineering

الهندسة العكسية لشفرة المصدر

• Take source code

• خذ كود المصدر

• Recover high level design information

• استرداد معلومات التصميم عالية المستوى

• By far the most widely performed type of reverse engineering

• إلى حد بعيد النوع الأكثر تنفيذًا على نطاق واسع من الهندسة العكسية

Reverse engineering objectives

Cope with complexity

- *Have a better understanding of voluminous and complex systems*
- *Extract relevant information and leave out low-level details*

Generate alternative views

- *Enable the designers to analyze the system from different angles*

Recover lost information

- *Changes made to the system are often undocumented;*
  - *This enlarges the gap between the design and the implementation*
- *Reverse engineering techniques retrieve the lost information*

Reverse engineering objectives

عكس الأهداف الهندسية

Cope with complexity

التعامل مع التعقيد

Have a better understanding of voluminous and complex systems Extract relevant information and leave out low-level details Generate alternative views

 الحصول على فهم أفضل للأنظمة الضخمة والمعقدة استخراج المعلومات ذات الصلة وترك التفاصيل ذات المستوى المنخفض إنشاء وجهات نظر بديلة

Enable the designers to analyze the system from different angles Recover lost information

 تمكين المصممين من تحليل النظام من زوايا مختلفة واستعادة المعلومات المفقودة

Changes made to the system are often undocumented;

غالبًا ما تكون التغييرات التي يتم إجراؤها على النظام غير موثقة؛

This enlarges the gap between the design and the implementation Reverse engineering techniques retrieve the lost information

يؤدي ذلك إلى توسيع الفجوة بين التصميم والتنفيذ. تعمل تقنيات الهندسة العكسية على استرداد المعلومات المفقودة

Reverse engineering objectives

CDetect side effects

- Detect problems due to the effect a change may have on the system before it results in failure

Facilitate reuse

- *Detect candidate system components that can be reused*

Reverse engineering objectives CDetect side effects

عكس الأهداف الهندسية كشف الآثار الجانبية

Detect problems due to the effect a change may have on the system before it results in failure

اكتشاف المشاكل بسبب تأثير التغيير على النظام قبل أن يؤدي إلى الفشل

Facilitate reuse

تسهيل إعادة الاستخدام

Detect candidate system components that can be reused

Detect مرشح مكونات النظام التي يمكن إعادة استخدامها

Source code reverse engineering techniques

- **Program analysis**
- **Program slicing**
- **Design recovery**
- **Architecture recovery**

Source code reverse engineering techniques

تقنيات الهندسة العكسية للكود المصدري

Program analysis

تحليل البرنامج

Program slicing Design recovery Architecture recovery

تقطيع البرنامج، استعادة التصميم، استعادة الهندسة المعمارية

Source code reverse engineering techniques

Program slicing

- concerned with analyzing all the statements in a program that may
  - Affect the value of a given variable at a certain point during execution
    - Looking backward
  - Be affected by the execution of a certain statement
    - Looking forward
- Can be either static or

Static slicing

- Considers all possible inputs
  - the resulting slices are usually quite large

Dynamic slicing

- Extracting parts of the program that contribute to the computation of the function according to a specific input

Source code reverse engineering techniques

تقنيات الهندسة العكسية للكود المصدري

Program slicing

تقطيع البرنامج

concerned with analyzing all the statements in a program that may o Affect the value of a given variable at a certain point during execution

تهتم بتحليل جميع العبارات الموجودة في البرنامج والتي قد تؤثر على قيمة متغير معين عند نقطة معينة أثناء التنفيذ

• Looking backward o Be affected by the execution of a certain statement

• النظر إلى الوراء o تتأثر بتنفيذ عبارة معينة

• Looking forward

• نتطلع

Can be either static or

يمكن أن تكون إما ثابتة أو

Static slicing

تقطيع ثابت

Considers all possible inputs o the resulting slices are usually quite large Dynamic slicing

يأخذ في الاعتبار جميع المدخلات الممكنة للشرائح الناتجة والتي عادة ما تكون تقطيعًا ديناميكيًا كبيرًا جدًا

Extracting parts of the program that contribute to the computation of the function according to a specific input

استخراج أجزاء من البرنامج تساهم في حساب الدالة وفق مدخلات محددة

Source code reverse engineering techniques

Design recovery

- Create design abstractions in order to understand what a program does, how it does it and why it does it.

- Examine multiple sources of knowledge including:
  - ❑ the system documentation (if available),
  - ❑ the knowledge that the software engineers have of the system

- Difficult to perform on large systems that have undergone ad-hoc maintenance for a long period of time
  - ❑ Documentation of such legacy systems is usually out of date
  - ❑ Original developers are usually no longer working within the organization

Source code reverse engineering techniques

تقنيات الهندسة العكسية للكود المصدري

Design recovery

استعادة التصميم

- Create design abstractions in order to understand what a program does, how it does it and why it does it.
- إنشاء تجريدات التصميم من أجل فهم ما يفعله البرنامج، وكيف يفعل ذلك، ولماذا يفعل ذلك.

- Examine multiple sources of knowledge including:
- دراسة مصادر المعرفة المتعددة بما في ذلك:

 the system documentation (if available),

 وثائق النظام (إن وجدت)،

 the knowledge that the software engineers have of the system

 المعرفة التي يمتلكها مهندسو البرمجيات بالنظام

Difficult to perform on large systems that have undergone ad-hoc maintenance for a long period of time

 من الصعب تنفيذه على الأنظمة الكبيرة التي خضعت لصيانة مخصصة لفترة طويلة من الزمن

 Documentation of such legacy systems is usually out of date

 عادةً ما يكون توثيق هذه الأنظمة القديمة قديمًا

 Original developers are usually no longer working within the organization

 عادةً ما يتوقف المطورون الأصليون عن العمل داخل المؤسسة

Source code reverse engineering techniques

Architecture recovery

- Aims to recover the overall system structure in terms of its high-level components and the way they interact

- There are several techniques
  - ❑ Using human experts
  - ❑ Recognizing known patterns
  - ❑ Static and dynamic analysis
  - ❑ Clustering and data mining

Source code reverse engineering techniques

تقنيات الهندسة العكسية للكود المصدري

Architecture recovery

استعادة العمارة

Aims to recover the overall system structure in terms of its high-level components and the way they interact

يهدف إلى استعادة بنية النظام الشاملة من حيث مكوناته عالية المستوى وطريقة تفاعلها

There are several techniques

هناك العديد من التقنيات

- Using human experts
- الاستعانة بالخبراء البشريين
- Recognizing known patterns
- التعرف على الأنماط المعروفة
- Static and dynamic analysis
- التحليل الساكن والديناميكي
- Clustering and data mining
- التجميع واستخراج البيانات

Reverse engineering tools

▪The process of reverse engineering is accomplished by making use of some tools that are categorized into:

- **Disassemblers** – A disassembler is used to convert binary code into assembly code and also used to extract strings, imported and exported functions, libraries etc. The disassemblers convert the machine language into a user-friendly format. There are different disassemblers that specialize in certain things.

- **Debuggers** – This tool expands the functionality of a disassembler by supporting the CPU registers, the hex duping of the program, view of stack etc. Using debuggers, the programmers can set breakpoints and edit the assembly code at run time. Debuggers analyze the binary in a similar way as the disassemblers and allow the reverser to step through the code by running one line at a time to investigate the results.

Reverse engineering tools

أدوات الهندسة العكسية

The process of reverse engineering is accomplished by making use of some tools that are categorized into:

تتم عملية الهندسة العكسية من خلال الاستفادة من بعض الأدوات التي تصنف إلى:

Disassemblers – A disassembler is used to convert binary code into assembly code and also used to extract strings, imported and exported functions, libraries etc.

المفككات - يتم استخدام المفكك لتحويل الكود الثنائي إلى كود التجميع ويستخدم أيضًا لاستخراج السلاسل والوظائف المستوردة والمصدرة والمكتبات وما إلى ذلك.

The disassemblers convert the machine language into a user-friendly format.

تقوم أدوات التفكيك بتحويل لغة الآلة إلى تنسيق سهل الاستخدام.

There are different disassemblers that specialize in certain things.

هناك مفككون مختلفون متخصصون في أشياء معينة.

 Debuggers – This tool expands the functionality of a disassembler by supporting the CPU registers, the hex duping of the program, view of stack etc.

 مصححات الأخطاء - تعمل هذه الأداة على توسيع وظائف المفكك من خلال دعم سجلات وحدة المعالجة المركزية، والخداع السداسي للبرنامج، وعرض المكدس وما إلى ذلك.

Using debuggers, the programmers can set breakpoints and edit the assembly code at run time.

باستخدام مصححات الأخطاء، يمكن للمبرمجين تعيين نقاط التوقف وتحرير كود التجميع في وقت التشغيل.

Debuggers analyze the binary in a similar way as the disassemblers and allow the reverser to step through the code by running one line at a time to investigate the results.

يقوم مصححو الأخطاء بتحليل الملف الثنائي بطريقة مشابهة لبرامج التفكيك ويسمحون للعاكس بالتنقل خلال التعليمات البرمجية عن طريق تشغيل سطر واحد في كل مرة للتحقق من النتائج.

Reverse engineering tools

- **Hex Editor**s – These editors allow the binary to be viewed in the editor and change it as per the requirements of the software. There are different types of hex editors available that are used for different functions.

- **PE and Resource Viewer** – The binary code is designed to run on a windows based machine and has a very specific data which tells how to set up and initialize a program. All the programs that run on windows should have a portable executable that supports the DLLs the program needs to borrow from.

Reverse engineering tools

أدوات الهندسة العكسية

 Hex Editors – These editors allow the binary to be viewed in the editor and change it as per the requirements of the software.

 المحررات السداسية - تسمح هذه المحررات بعرض الملف الثنائي في المحرر وتغييره وفقًا لمتطلبات البرنامج.

There are different types of hex editors available that are used for different functions.

هناك أنواع مختلفة من المحررات السداسية المتاحة والتي يتم استخدامها لوظائف مختلفة.

 PE and Resource Viewer – The binary code is designed to run on a windows based machine and has a very specific data which tells how to set up and initialize a program.

 PE و Resource Viewer - تم تصميم الكود الثنائي ليعمل على جهاز يعمل بنظام Windows ويحتوي على بيانات محددة للغاية توضح كيفية إعداد البرنامج وتهيئته.

All the programs that run on windows should have a portable executable that supports the DLLs the program needs to borrow from.

يجب أن تحتوي جميع البرامج التي تعمل على نظام التشغيل Windows على ملف قابل للتنفيذ محمول يدعم ملفات DLL التي يحتاج البرنامج إلى الاقتراض منها.