1. 網路名詞
2. 運用Ack封包來告訴傳送方已收到封包的同時也傳送對方所需的資料
3. 以IP位置查詢為對應的MAC位置
4. 傳輸資料不需進行同步，可隨時傳送以最大化傳輸效率
5. 是將頻段分割成256個各具有不同載波信號的子頻道，再將數位的資料利用QAM調變方式，分配調變於256個載波信號的子頻道上，每個頻道傳輸的資料量可以依傳輸線路的干擾與串音的情況來調整。
6. 送出資料的同時也監聽傳輸訊號，若與傳輸之內容不同即可能發生碰撞
7. 將子網路中的master節點作為另一個網路的slave節點以連結兩網路
8. 紀錄其他工作站來需要多久的時間傳送訊框
9. 非對稱是加密所需相較於對稱式加密的計算量大，會降低傳輸效率
10. kernel
11. 單核心 : 被視作為運行在單一位址空間的單一的行程，核心提供的所有服務，都以特權模式，在這個大型的核心位址空間中運作

優點 → 設計簡單、功能完善、執行效率高

缺點 → 移植性不佳、安全性較差

1. 微核心 : 將核心功能模組化，劃分成幾個獨立的行程，各自運行，這些程序被稱為服務（service），並運行在不同的位址空間，只有需要絕對特權的程序，才能在具特權的執行模式下運行，其餘的程序則在使用者空間運行

優點 → 移植性高、穩定性高

缺點 → 效能較差

1. heap overflow attack
2. 對電腦輸入大量資料，多到超過原訂的資料記憶體配置空間（稱為資料緩衝區）。於是，超出空間的溢出資料改為覆寫到heap位置、程式區段位置上，導致應用程式不正常關閉或被駭客埋設惡意程式，並驅使系統跳躍到惡意程式段去執行惡意程式，進而開啟後門或取得系統主導權
3. 加上輸入值與緩衝區長度檢查機制、使用工具檢查程式是否有安全性漏洞
4. 可行性

→ 用來評估資訊系統發展計畫是否有辦法進行

→ 經濟可行性、分險分析、作業可行性、技術可行性

1. 物件導向設計

→ 結構化分析與設計法注重於資料流，其彈性較差且耗時，而物件導向則以現實世界中的實體作為物件刀做分析對象，可以清楚資料實體間的互動關係

→ 將實體中具有相似屬性與方法的物件抽象化為類別，並以此類別產生系統物件，描述與其他類別間的互動關係

1. A、BC、BD、CD
2. 方法覆寫與重載

多載 → 指多個名稱相同但參數個數或型別不同的方法，編譯器依傳入參數的個數、型別與順序決定使用哪一個方法

重載 → 改寫繼承來自父類別的方法或式介面的方法中的內容

1. 程式設計
2. 讀取陣列中的值並相加成得X，同時找到陣列中最大值M與最小值S
3. 如果缺少的值為最前或是最後的數字，(M-S+1) < N → 回傳S - 1 或是 M + 1
4. 缺少的非頭尾值，則計算N-1之所有值總和Z，Z - X之值即為缺少的值

int num(int a[], int n){

int s = 0, m = MAX\_INT, sum = 0;

for(int i = 0; i < n; i++){

if(a[i] > m) m = a[i];

if(a[i] < s) s = a[i];

sum = sum + a[i];

}

if(m-s+1 < n) return m+1;

else return (m+s)\*n/2 - sum;

}