

HỌC VIỆN KỸ THUẬT QUÂN SỰ
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN KHOA HỌC MÁY TÍNH

HƯỚNG DẪN LÀM BÀI TẬP MÔN HỌC

“LÝ THUYẾT HỆ ĐIỀU HÀNH”
“Operating System Concepts” - Silberschatz

Giáo viên: GVC. TS. Hà Chí Trung

Điện thoại: 0385582102

Email: hct2009@yahoo.com

<https://www.facebook.com/groups/805527496281567/>

HÀ NỘI, 11 - 2018

I. Giải quyết bài tập môn

Các em sinh viên Khóa 15+ thân mến!

Để trở thành một sinh viên công nghệ thông tin tốt, một lập trình viên giỏi thì mỗi sinh viên nên và cần thiết thực hiện các project cá nhân, tốt nhất là ở tất cả các môn học trong suốt quá trình học tập. Tài liệu này nhằm mục đích khuyến khích và tạo điều kiện cho sinh viên phát triển các kỹ năng: kỹ năng vận dụng lý thuyết; kỹ năng lập trình; kỹ năng viết báo cáo; kỹ năng trình bày báo cáo; kỹ năng làm việc nhóm; kỹ năng giải quyết vấn đề mới.

Các đề tài đều có tính mở cao. Có nghĩa là, sinh viên có thể thực hiện ở mức tối thiểu (đạt điểm tối thiểu) hoặc phát triển các chức năng đầy đủ hoàn thiện, tùy vào mức độ phức tạp, chi tiết khi xây dựng chương trình hay tìm hiểu lý thuyết để nâng cao điểm số đánh giá.

I.1. Xây dựng chương trình

Bài tập môn Lý thuyết hệ điều hành có 3 dạng: lý thuyết, bài tập lập trình và bài tập tự chọn:

- Lý thuyết: tìm hiểu một vấn đề lý thuyết liên quan đến hệ điều hành, nắm rõ lý thuyết để thuyết trình trước lớp và chuẩn bị báo cáo và chương trình demo hoặc demo các kỹ năng thực hành trên máy tính liên quan đến lý thuyết đó;
- Bài tập lập trình: tìm hiểu một vấn đề lý thuyết và viết chương trình demo trên bất cứ nền tảng OS nào (như Windows 7, 8, 10, DOS, ubuntu, linux...) với bất cứ ngôn ngữ nào (C, C++, Visual C++, C#.net, Delphi, Visual Basic, Visual Basic.net, Prolog, Lisp, Java, Python...) Chương trình cần có giao diện đẹp và dễ sử dụng. Phần mã lệnh trong chương trình cần chú giải rõ ràng các lớp, các hàm, các identifiers chính được sử dụng và có hướng dẫn sử dụng đầy đủ. Đối với các bài tập là viết trò chơi hay minh họa thuật toán phải thể hiện bằng đồ họa, hình vẽ trên màn hình giao diện, mô phỏng quá trình làm việc của thuật toán (và / hoặc chỉ rõ các bước biến đổi dữ liệu và không gian trạng thái của bài toán);
- Tự chọn: sinh viên có thể tự đưa ra đề tài và thống nhất với giáo viên, đề tài tự chọn phải không trùng với các đề tài trong danh sách (bên dưới).

I.2. Viết tiểu luận báo cáo

Báo cáo yêu cầu trình bày đầy đủ và ngắn gọn. **Với đề tài lý thuyết cần chuẩn bị cả file Word (.doc, docx, không quá 20 trang A4, Time New Romans, cỡ chữ 14,**

giãn dòng 1.3pt, lề: trên dưới 2cm, phải 3cm, trái 2cm, thụt đầu dòng 1cm) và **file báo cáo trên PowerPoint** (không quá 20 trang slides); **với bài tập lập trình chỉ viết báo cáo trên PowerPoint** (không quá 20 trang slides). Báo cáo phải thể hiện được các phần sau:

1. **Trang mở đầu:** thể hiện tên trường, tên đề tài, người hướng dẫn, tên những người thực hiện, mã lớp, ngày hoàn thành (1 slide).
2. **Mục lục:** thể hiện tên các đề mục tương tự như các phần ở dưới đây (1 A4 và/hoặc 1 slide).
3. **Đặt vấn đề:** mục đích của việc giải quyết bài tập được giao. Nêu và phân tích các vấn đề cần giải quyết của đề tài được giao (1 A4 và/hoặc 1 slide).
4. **Phân công công việc:** với đề tài nhóm, cần chỉ rõ từng thành viên trong nhóm thực hiện công việc gì trong việc giải quyết đề tài?
5. **Cơ sở lý thuyết:** Nêu các kiến thức lý thuyết liên quan trực tiếp đến đề tài được giao (ví dụ thể nào là lập lịch cho tiến trình, các thuật toán lập lịch cho tiến trình v.v.).
6. **Sơ đồ công việc:** Nếu trong báo cáo có sử dụng thuật toán cần thể hiện thuật toán đó bằng sơ đồ khối công việc, hoặc sơ đồ thuật toán làm việc của chương trình (trình bày ngắn gọn).
7. **Phát triển chương trình:** Nêu rõ cấu trúc của chương trình, ngôn ngữ được sử dụng, các moduls, các class hay cấu trúc được xây dựng, các hàm cơ bản được sử dụng (thể hiện ngắn gọn, nêu rõ ý nghĩa của các hàm, đầu vào, đầu ra – không quá 2 trang A4 và/hoặc 2 slides).
8. **Giao diện người dùng:** Thể hiện giao diện chính của chương trình, màn hình nhập dữ liệu ban đầu và màn hình kết quả trả về (khoảng 2 slides).
9. **Phụ lục:** Toàn bộ hoặc 1 phần demo code chính của chương trình có chú giải rõ ràng (không quá 3 trang A4 và/hoặc 3 slides).
10. **Tài liệu tham khảo** (nếu có, bao gồm các sách được tham khảo, website, các code mã nguồn mở được sử dụng (toàn bộ không quá 1 trang A4 và/hoặc 1 slide).

Tổng cộng tiêu luận báo cáo trên Word không quá 20 trang A4. File trình chiếu trên PowerPoint không quá 20 slides, không ít hơn 10 slides.

II. Trả bài tập môn

- Khi báo cáo, sinh viên cần mang theo **mã nguồn chương trình đã viết, mã đã được biên dịch (.exe, .a), tiểu luận báo cáo, file trình chiếu**, máy tính cá nhân (nếu có và nếu cần thiết cho chương trình, hoặc copy đem theo trên USB);
- Sinh viên có thể trả trước ngày quy định (trả chậm mỗi 1 ngày so với quy định trừ 0.5 điểm, quá 1 tuần không được chấp nhận). khi đó có thể thông qua lớp trưởng hoặc tự liên hệ với giáo viên *tại Phòng thí nghiệm Multimedia và tính toán thông minh, Khoa CNTT, tại H2-307* để giáo viên tiếp nhận bài tập.

Báo cáo được chấp nhận khi và chỉ khi chương trình hoạt động được và có kết quả, có đầy đủ nội dung theo yêu cầu. Cấu trúc thư mục bài tập môn như bảng dưới (sinh viên cần tuân thủ bắt buộc – sai quy cách trừ 1 điểm):

Quy cách đặt tên	Giải thích
Thư mục gốc: <i>L6-NVTuấn-VQAnh</i>	Tên thư mục gốc phải thể hiện được mã số đề tài, tên vắn tắt của các thành viên trong nhóm.
Các files và thư mục con (nếu có)	
Source	Chứa mã nguồn và các file thi hành của chương trình. Có thể tạo file cài đặt nếu chương trình đòi hỏi các thư viện khác. Yêu cầu phải chạy được chương trình chỉ với các phần ở thư mục này.
File báo cáo trên Word (với các đề tài lý thuyết) <i>L6-NVTuấn-VQAnh.docx</i>	Sinh viên đồng thời tạo ra 1 bản pdf của file này. Tên file cần được đặt giống tên thư mục gốc. VD: <i>L6-NVTuấn-VQAnh.pdf</i>
File báo cáo PowerPoint Ví dụ: <i>L6-NVTuấn-VQAnh.ppts</i>	Chứa file báo cáo trình bày theo đúng quy định ở trên. File báo cáo được đặt giống tên thư mục gốc. Slile cần ít chữ, chủ yếu sử dụng sơ đồ, hình vẽ
Reference	Chứa các tài liệu tham khảo (nếu có)
readme.txt (unicode) hoặc các file hướng dẫn sử dụng đặt tên là guide.*	Chứa thông tin chi tiết của các thành viên nhóm thực hiện (Họ tên đầy đủ, ngày tháng năm sinh, lớp khóa). Các tập tin hướng dẫn sử dụng nếu có (format trình bày tùy ý).
Chú ý: Toàn bộ thư mục sau đó nén thành file .zip, .rar, .7z hay .taz..., tên file trùng với tên thư mục để lưu trữ	

Cách tính điểm phần bài tập lập trình:

1. Chương trình làm việc tốt, thuật toán được xây dựng đúng đắn – max 3 điểm.
2. Giao diện dễ sử dụng, thân thiện và thể hiện tốt bố cục cũng như phần đồ họa – max 3 điểm.
3. Code chương trình được viết có cấu trúc mạch lạc, rõ ràng, thể hiện chú thích miêu tả - 1 điểm.
4. Slide thể hiện rõ ràng, sạch sẽ, bố cục hợp lý, thể hiện được các công thức (nếu có), hình vẽ (sơ đồ làm việc), hình minh họa – 2 điểm.
5. Kỹ năng trình bày - 1 điểm.
6. Trình bày báo cáo không thể hiện được kiến thức về những vấn đề liên quan, không nắm rõ về nội dung bài tập được giao, không làm chủ được chương trình, không giải thích được những khái niệm được sử dụng trong báo cáo cũng như trong chương trình... thì không có điểm số cho mục 1, 2.

Tổng cộng: không quá 10 điểm.

Cách tính điểm phần bài tập lý thuyết:

1. Tiểu luận rõ ràng, sạch sẽ, bố cục hợp lý theo quy định, thể hiện được các công thức (nếu có), hình vẽ (sơ đồ làm việc), hình minh họa – 3 điểm;
2. Mức độ bao phủ nội dung so với yêu cầu (max 4 điểm)
3. Kỹ năng trình bày - 2 điểm.
4. Trình bày báo cáo không thể hiện được kiến thức về những vấn đề liên quan, không nắm rõ về nội dung bài tập được giao, không giải thích được những khái niệm được sử dụng trong báo cáo cũng như trong chương trình... thì không có điểm số cho mục 2.

Tổng cộng: không quá 10 điểm.

Lưu ý:

- Mỗi nhóm có không quá 2 sinh viên (có thể gồm 1 hoặc 2 sinh viên);
- Các đề tài là bắt buộc, sinh viên cần chủ động tìm tài liệu và viết báo cáo, không chấp nhận thay đổi đề tài;
- Sinh viên 2 lớp khác nhau, nếu trùng mã đề tài thì báo cáo và chương trình phải khác nhau, nếu trùng, lấy điểm cao nhất chia đều cho các nhóm;
- Mọi thắc mắc trao đổi để lại comment trong nhóm Operating System Concepts
- **Attention! Không kéo dài thời gian làm bài tập, tập trung xử lý trong 1 tuần. Các nhóm làm bài tập lý thuyết trả trước ngày 23/11/2018; các nhóm làm bài tập lập trình trả trước 25/11/2018. Trả trước 20/11/2018 được xem xét cộng điểm.**

DANH SÁCH CÁC ĐỀ BÀI TẬP

BÀI TẬP LẬP TRÌNH

TT	Nội dung	Tham khảo		
	BÀI TẬP VỚI TIẾN TRÌNH VÀ LUỒNG			
B1.	Lập lịch cho CPU Viết chương trình mô phỏng các giải thuật định thời: First Come First Served (FCFS), Round Robbin (RR), Shortest Job First (SJF), Shortest Remain Time (SRT). Tính toán các thông số và đánh giá độ hiệu quả của từng giải thuật. Vẽ biểu đồ Gantt với các thông số, mô phỏng từng bước.	(cần tính waiting time của mỗi process; average WT; turnaround time; average TaT; Response time, average RT; throughput). Vẽ biểu đồ Gantt với các thông số.		
B2.	MyTaskManager Viết chương trình thực hiện một số chức năng tương tự như phần quản lý tiến trình trong Task Manager: thực thi các chương trình ngoài (tiến trình), kết thúc chương trình, hiển thị các thông tin liên quan đến tiến trình đó.	Sử dụng các hàm API		
B3.	Banker's Algorithm Viết chương trình mô phỏng giải thuật nhà băng (banker's algorithm) của Dijkstra để tránh deadlock. Số lượng các nguồn tài nguyên, các yêu cầu của hệ thống được tạo ngẫu nhiên. Hãy hiện thực cách xét trạng thái hiện thời của hệ thống là an toàn hay không an toàn. Nếu trạng thái là an toàn, chương trình phải chỉ ra cách để thu hồi tài nguyên hệ thống. Với mỗi yêu cầu cung cấp tài nguyên, hãy hiển thị ra màn hình cách cấp phát, thu hồi tương ứng. Giao diện Winform.	Chapter 7. Deadlock		
B4.	Bakery's Algorithm	Chapter 5. Synchronization		

	Viết chương trình mô phỏng giải thuật Bakery cho phép đồng bộ hóa tiến trình. Giao diện Winform			
B5.	<p>Giao tiếp giữa 2 tiến trình</p> <p>Tạo ra 2 Tiến trình. Tiến trình thứ nhất đọc từ file nhiều chuỗi liên tiếp, mỗi chuỗi gồm các phép toán +, -, *, / và hai toán hạng.</p> <p>Ví dụ trong file sẽ lưu các chuỗi dạng là các phép toán hai ngôi, ví dụ:</p> <p>2 + 3 1 - 2 4 * 6 15 / 3</p> <p>Sau đó Tiến trình thứ nhất truyền các chuỗi dữ liệu này cho Tiến trình thứ hai. Tiến trình thứ hai thực hiện tính toán và trả chuỗi kết quả về lại cho Tiến trình đầu tiên để ghi lại vào file như sau:</p> <p>2 + 3 = 5 1 - 2 = -1 4 * 6 = 24 15 / 3 = 5</p> <p>Sử dụng một, hoặc cả 3 cơ chế giao tiếp giữa các tiến trình: Message Queue; Shared Memory; Pipe.</p>			
B6.	<p>Tiến trình cha con</p> <p>Viết chương trình tạo ra một Tiến trình con. Tiến trình cha sẽ sinh ra ngẫu nhiên k files ($k < 10$), mỗi file chứa một dãy ngẫu nhiên N số nguyên ($N \leq 5000$). Tiến trình con sẽ đọc dữ liệu từ các file này, sắp xếp chúng theo thứ tự tăng dần và ghi vào một file khác tên là output.txt. Gợi ý sử dụng phương pháp MergeSort kết hợp Quicksort.</p>			


B7.	<p>Giao tiếp giữa 2 tiến trình</p> <p>Viết chương trình gồm có 2 tiến trình. Tiến trình thứ nhất cho người dùng nhập vào từ bàn phím một chuỗi biểu diễn một phép tính gồm các phần tử +, -, (,). Độ ưu tiên của các phép tính trong ngoặc (cặp dấu (và)) là cao nhất, phép + và – cùng độ ưu tiên. Ví dụ: $1+2+(2-3-4) - ((3+4)-5)$ $(1+(-2))-((3+4)-5))$</p> <p>Sau đó truyền chuỗi dữ liệu này cho tiến trình thứ hai. Tiến trình thứ hai thực hiện tính toán trên và trả kết quả về cho tiến trình thứ nhất để hiển thị cho người sử dụng biết.</p> <p>Sử dụng một, hoặc cả 3 cơ chế giao tiếp giữa các tiến trình: Message Queue; Shared Memory; Pipe.</p>	Tìm hiểu thêm giải thuật tính giá trị biểu thức số đã học trong môn cấu trúc dữ liệu và giải thuật, sử dụng stack kết hợp queue		
B8.	<p>Giao tiếp giữa 2 tiến trình</p> <p>Tạo ra 2 tiến trình P0 và P1. Tiến trình P0 đọc từ file có chứa K ($K > 1$) dãy số nguyên liên tiếp (mỗi dãy có thể có số phần tử khác nhau). Sau đó quá trình này gửi lần lượt các dãy này cho tiến trình P1. P1 thực hiện sắp xếp mỗi dãy theo thứ tự tăng dần, đồng thời tính tổng của dãy đó và gửi dãy kết quả cùng với tổng tính được tương ứng cho mỗi dãy về lại tiến trình P0. Khi này, P0 thực hiện ghi các dãy kết quả vào lại file sao cho: dãy có tổng nhỏ nhất sẽ được ghi đầu tiên, dãy có tổng lớn hơn sẽ được ghi sau.</p> <p>Sử dụng một, hoặc cả 3 cơ chế giao tiếp giữa các tiến trình: Message Queue; Shared Memory; Pipe.</p>			
B9.	<p>Giao tiếp giữa 2 tiến trình</p> <p>Tạo ra 2 tiến trình P0 và P1. Quá trình P0 đọc từ file một ma trận vuông cấp $N \times N$, sau đó gửi ma trận này cho P1. Tiến trình P1 sẽ thực hiện nghịch đảo ma trận này và ghi</p>			

	<p>kết quả xuống một file khác. Nếu ma trận không thể nghịch đảo được, có thể tìm một ma trận giả nghịch đảo Moore-Penrose, hoặc ghi ra file kết quả dòng thông báo tương ứng. Sử dụng một, hoặc cả 3 cơ chế giao tiếp giữa các tiến trình: Message Queue; Shared Memory; Pipe.</p>			
B10.	<p>Producers-consumers problem Sử dụng N thread ($N > 1$) và các phương thức đồng bộ trên thread để viết chương trình producers-consumers dùng chung 1 bounded buffer với trường hợp có 1 producer và nhiều consumer.</p>			
B11.	<p>Producers-consumers problem Sử dụng N thread ($N > 1$) và các phương thức đồng bộ trên thread để viết chương trình producers-consumers dùng chung 1 bounded buffer trong trường hợp có nhiều producer và nhiều consumer.</p>			
B12.	<p>Dining philosophers problem Viết chương trình giải quyết bài toán 5 triết gia ăn tối. Chương trình phải tạo ra 5 threads mô phỏng hoạt động của 5 triết gia. Dùng semaphore để đồng bộ hoạt động của 5 triết gia này.</p>			
B13.	<p>Reader Writer problem Viết chương trình giải quyết bài toán reader/writer dạng tổng quát. Chương trình phải tạo ra 5 threads đồng thời với hoạt động đọc/ghi file bất kỳ để thử nghiệm giải thuật.</p>			
B14.	<p>Multithread Ứng dụng multithread trong bài toán nhân ma trận. Đọc vào 2 ma trận A & B từ hai file riêng biệt, sau đó dùng</p>			

	<p>nhiều thread để thực hiện việc nhân 2 ma trận này. Ghi ma trận kết quả ra một file khác.</p>			
B15.	<p>Multithread Đọc vào một ma trận A từ file, sau đó dùng N threads ($N > 1$) để thực hiện đồng thời phép biến đổi trên ma trận này như sau: tạo ra ma trận mới với mỗi phần tử trên ma trận là trung bình cộng của các phần tử chung quanh nó. Ghi ma trận kết quả ra một file khác.</p>			
B16.	<p>Multithread Đọc vào một ma trận (kích thước $N \times N$) từ file, sau đó dùng N threads ($N > 1$) để thực hiện đồng thời việc tính tổng của từng hàng trên ma trận (1 thread nếu tính tổng một hàng xong có thể tính tổng của một hàng khác). Dùng một thread để thu thập các kết quả của các thread kia và ghi vào một file kết quả.</p>			
B17.	<p>Multithread Ứng dụng multithread trong bài toán sau: Đọc vào một ma trận A từ file và nhập vào một số k, sau đó dùng ít nhất 2 thread để thực hiện đồng thời việc tìm kiếm trên các ma trận này xem có bao nhiêu phần tử giống k. Mỗi lần tìm thấy phần tử giống k thì tăng biến chung number (khởi động là 0) lên 1.</p>			
B18.	<p>Multithread Viết chương trình chuyển đổi một biểu thức từ dạng trung tố sang các dạng tiền tố, hậu tố và ngược lại, mỗi một công việc được thực hiện trên một luồng riêng biệt</p>	Tham khảo cấu trúc dữ liệu và giải thuật.		
B19.	<p>Multithread with MergeSort</p>			

	Thuật toán sắp xếp trộn với files ngoài (Merge Sort): Ứng dụng multithread trong bài toán sau : Dùng 1 hoặc nhiều thread để đọc dữ liệu các số nguyên từ 1 hay nhiều files ngoài, tương tự dùng N threads để sắp xếp dãy theo thứ tự tăng dần. Mỗi thread lấy một phần dãy a và sắp theo thứ tự tăng dần. Sau đó, sử dụng 1 thread trộn các dãy do N threads vừa rồi đã sắp xếp thành dãy kết quả.			
B20.	Multithread with Eratosthenes algorithm Cho người dùng nhập vào 2 số nguyên dương a và b ($a, b < 10.000.000.000$), sau đó dùng N threads để tìm kiếm các số nguyên tố nằm trong khoảng 2 số đã nhập. Mỗi lần tìm được một số thì sẽ tăng biến chung number (khởi động là 0) lên 1. Ghi kết quả ra file output.txt, dòng đầu tiên là số số nguyên tố tìm được, dòng tiếp theo là các số nguyên tố trong khoảng $[a, b]$ cách nhau bởi dấu tab.			
B21.	Memory-Resident Program Viết chương trình thường trú thực hiện chức năng nhắc lịch (lịch học thời khóa biểu, hẹn giờ, v.v.)			
B22.	Mirror Viết ứng dụng sử dụng 2 tiến trình (hoặc luồng), trên một cửa sổ di chuột để vẽ các đường tùy ý, trên cửa sổ còn lại sẽ vẽ theo chiều ngược lại (đối xứng qua gương).			
B23.	Follow Me! Viết chương trình có 2 cửa sổ, trên một cửa sổ này thực hiện thao tác kéo thả một hay một số đối tượng (hình tròn, vuông, v.v.) trên cửa sổ còn lại đối tượng tương ứng di chuyển theo.			

	LẬP TRÌNH SOCKET			
B24.	Server-Clients Tìm hiểu về lập trình Socket: Viết chương trình tạo hai quá trình server và client ở hai máy khác nhau. Client nhận một dãy số từ người sử dụng và gửi cho quá trình server. Server sắp xếp chuỗi này theo thứ tự tăng dần và gửi trả lại client để hiển thị cho người dùng biết.			
B25.	Server-Clients Tìm hiểu về lập trình Socket: Tạo hai quá trình server và client ở hai máy khác nhau. Client nhận một số nguyên dương từ người sử dụng và gửi cho quá trình server. Server phân tích số này thành thừa số nguyên tố và gửi trả lại client để hiển thị cho người dùng biết.			
B26.	Server-Client Chatbox Dùng socket để viết chương trình chat giữa hai quá trình trên hai máy khác nhau. Gợi ý : viết 1 chương trình chat client (để người dùng sử dụng) và dùng một chương trình chat server để quản lý các chat client đó.			
B27.	Server-Client Messages để lập trình ứng dụng tcpServer và tcpClient với những tính năng sau: 1. Thu nhỏ về System Tray và bật lên khi có thông điệp đến 2. Giao diện có dạng tương tự như sau:			

				
B28.	Server-Client Chatbox Giao tiếp giữa các máy tính (chat nhóm): Viết 2 chương trình listen và send. Chương trình listen dùng để nhận thông tin từ các người khác gửi đến và hiển thị lên màn hình. Chương trình send cho phép gửi thông tin đến tất cả các người chạy chương trình listen trên mạng. Dùng cơ chế broadcast.			
B29.	Server-Client Chatbox Giao tiếp giữa các máy tính (chat nhóm): Viết 2 chương trình listen và send. Chương trình listen dùng để nhận thông tin từ các người khác gửi đến và hiển thị lên màn hình. Chương trình send cho phép gửi thông tin đến tất cả các người chạy chương trình listen trên mạng. Dùng cơ chế multicast.			
	BÀI TẬP QUẢN LÝ BỘ NHỚ			
B30.	Virtual Memory Viết chương trình mô phỏng các giải thuật thay thế trang: OPT, FIFO, LRU, LFU, NUR, Second Chance. Input: số khung trang, chuỗi tham khảo trang, yêu cầu thay thế trang...được tạo ngẫu nhiên.			

	Output: sơ đồ các bước thay thế trang, số PAGE FAULTS. Đánh giá độ hiệu quả (so sánh số page fault) của từng giải thuật.			
	LÀM VIỆC VỚI TẬP TIN VÀ CÁC THIẾT BỊ LƯU TRỮ			
B31.	Viết chương trình cho phép nhập vào địa chỉ logic hoặc địa chỉ vật lý của 1 sector trên ổ đĩa. Đọc và hiển thị ra màn hình dưới dạng Hex nội dung của sector đó. Lưu nội dung đó vào một file.			
B32.	Viết chương trình count để đếm số lần xuất hiện của một chuỗi trong các file text (ASCII) được liệt kê trong dòng lệnh, không phân biệt chữ thường và chữ hoa. Chương trình sau khi dịch sẽ được chạy bằng dòng lệnh: count <chuỗi> <tên file> ... Ví dụ: count yes file1.txt file2.txt <i>(đếm chuỗi yes trong file1.txt và file2.txt)</i> Chương trình cần sử dụng nhiều tiến trình đồng hành. Số lượng tiến trình bằng với số file text trên dòng lệnh. Chương trình tạo ra nhiều tiến trình con bằng cách gọi chương trình count để xử lý các file tương ứng. Theo dõi thời gian chạy và xuất ra màn hình.			
B33.	Disk Scheduling Viết chương trình mô phỏng các thuật toán Lập lịch cho đầu từ: FCFS, SSTF, SCAN, C-SCAN, C-LOOK. Giao diện Winform, đồ họa.			
B34.	File Reader			

	Viết chương trình cho phép đọc, ghi nội dung file text theo từng sector một. Hoặc đọc nội dung ở các sector được chỉ định. Giao diện Winform.			
B35.	My Folder Explorer Viết chương trình tương tự Windows Explorer (ví dụ trên lớp)			
B36.	My Commander Viết chương trình thực hiện một số chức năng tương tự TotalComander.	https://www.ghisler.com/		
B37.	My Registry Viết chương trình cho phép làm việc với sổ đăng ký của HĐH Windows (tham khảo lệnh regedit từ cửa sổ run)			
B38.	Coder-Decoder Viết chương trình mã hóa nội dung của một tập tin & chương trình giải mã (bằng thuật toán mã hóa và giải mã bất kỳ) tương ứng. Giao tiếp qua dòng lệnh trong Shell (command prompt). Ví dụ: code.exe file1.txt output.xxx và decode.exe output.xxx file2.txt Chương trình code.exe sẽ nén file1.txt vào file output.xxx và decode.exe sẽ giải nén file output.xxx vào file2.txt			
B39.	My-DOS Viết chương trình giả lập các lệnh của DOS: MD, CD, RD, DEL, REN, TYPE, COPY, COPY CON, MOVE, DIR bằng cách sử dụng các hàm của C.	(tham khảo các chương trình như Total Commander)		
B40.	Memory Allocation Viết chương trình mô phỏng các giải thuật cấp phát bộ nhớ first-fit, best-fit, và worst-fit theo từng bước, kích			

	thước bộ nhớ được nhập vào, tình trạng bộ nhớ ban đầu được sinh ngẫu nhiên, số tiến trình và kích cỡ tiến trình được sinh ngẫu nhiên. Giao diện đồ họa.			
B41.	File Copy Viết chương trình thực hiện chức năng copy các file văn bản kích thước lớn: Input: 1 file văn bản kích thước từ 100MB đến 10GB Output: bản sao của file Input - Sử dụng 2 process và một buffer dùng chung (kích thước buffer < 5 KB): Process 1 thực hiện chức năng đọc dữ liệu từ file Input vào buffer; Process 2 thực hiện chức năng ghi dữ liệu từ buffer ra file Output - Giải quyết vấn đề tranh chấp bằng 2 cơ chế: busy-waiting và sleep-and-wakeup. Nhận xét và so sánh thời gian chạy của 2 cơ chế này.			
MỘT SỐ BÀI TẬP KHÁC				
B42.	Emails Management Viết ứng dụng quản lý, gửi nhận Emails, đọc Emails.	http://csharp.net-informations.com/communications/csharp-communications-tutorial.htm		
B43.	Chatbox Ứng dụng chat trên C#, cho phép gửi nhận message, hình ảnh và dữ liệu khác.	http://csharp.net-informations.com/communications/csharp-chat-server-programming.htm		
B44.	C# Multi threaded Server Socket programming Xem thêm tại:	http://csharp.net-informations.com/communications/csharp-multi-threaded-socket.htm https://yinyangit.wordpress.com/2011/07/24/csharp-simple-example-multiconnection-tcp-server/		
B45.	Game Tetris	Chương 3. Stefan Björnander-C ++ Windows Programming-Packt Publishing (2016).pdf		

	Viết trò chơi xử lý các sự kiện bàn phím cho trò chơi Tetris			
B46.	Word Processor Viết chương trình soạn thảo văn bản với các chức năng cơ bản. Tham khảo: Chương 6. Stefan Björnander-C ++ Windows Programming-Packt Publishing (2016).pdf	Chương 6. Stefan Björnander-C ++ Windows Programming-Packt Publishing (2016).pdf		
B47.	MyExcel Viết chương trình thực hiện một số chức năng của bảng tính tương tự Excel Tham khảo: Chương 8. Building a Spreadsheet Application. Stefan Björnander-C ++ Windows Programming-Packt Publishing (2016).pdf	Chương 8. Building a Spreadsheet Application. Stefan Björnander-C ++ Windows Programming-Packt Publishing (2016).pdf		

BÀI TẬP LÝ THUYẾT

TT	Nội dung	Tham khảo
L1.	Tìm hiểu và trình bày về kiến trúc và hoạt động hệ điều hành Android, làm rõ các vấn đề: quản lý tiến trình, quản lý bộ nhớ, quản lý lưu trữ.	
L2.	Tìm hiểu và trình bày về kiến trúc và hoạt động hệ điều hành iOS, làm rõ các vấn đề: quản lý tiến trình, quản lý bộ nhớ, quản lý lưu trữ.	
L3.	Tìm hiểu cơ chế boot và cách cài đặt multiboot cho nhiều hệ điều hành trên một máy đơn.	
L4.	Tìm hiểu và trình bày về cấu trúc file EXE, file COM và cơ chế đính kèm của virus vào các loại file này.	
L5.	Tìm hiểu các cơ chế lây nhiễm của Malware.	
L6.	Tìm hiểu và trình bày về kiến trúc của Solid-state drive (SSD) và các hệ thống quản lý file trên SSD	
L7.	Tìm hiểu và trình bày về nguyên lý, cấu tạo và hoạt động của các tường lửa của hệ điều hành họ Windows, cấu hình tường lửa cho một hệ điều hành Windows cụ thể.	

L8.	Tìm hiểu và trình bày về cơ chế làm việc của hệ thống quản lý File FAT 32 trong các hệ điều hành Windows	
L9.	Tìm hiểu và trình bày về cơ chế làm việc của hệ thống quản lý File NTFS trong các hệ điều hành Windows.	
L10.	Tìm hiểu và trình bày về cơ chế làm việc của hệ thống quản lý File EXT3 hoặc EXT4 trong các hệ điều hành họ Linux.	
L11.	Tìm hiểu và trình bày về hệ thống quản lý file trong hệ điều hành Mac OS (Apple File System - APFS).	
L12.	Tìm hiểu và trình bày về hệ thống quản lý file trong máy ảo VMware VMFS5 hoặc VMFS 6 (Virtual Machine File System)	
L13.	Tìm hiểu và trình bày về nguyên lý, cấu tạo, hoạt động của các Máy ảo (đọc, dịch và báo cáo tóm tắt)	Chapter 16. Virtual Machines
L14.	Tìm hiểu và trình bày về cơ chế bảo vệ trong các hệ điều hành (đọc, dịch và báo cáo tóm tắt)	Chapter 14. Protection
L15.	Tìm hiểu và trình bày về cơ chế bảo mật trong các hệ điều hành (đọc, dịch và báo cáo tóm tắt)	Chapter 15. Security
L16.	Tìm hiểu về ngắt (interrupt), cơ chế ngắt và trình phục vụ ngắt trong hệ điều hành Windows	
L17.	Tìm hiểu về cơ chế, các thuật toán, cách thức làm việc và cấu hình Firewall trong một hệ điều hành họ Unix/Linux.	
L18.	Tìm hiểu và trình bày về cơ chế làm việc của hệ thống quản lý File HPFS (High Performance File System)	
L19.	Tìm hiểu về Peer-2-Peer File Sharing (Minh họa bằng chương trình hoặc cài đặt chức năng File Sharing của Windows)	
L20.	Tìm hiểu và làm rõ các thành phần của Grid Computing, làm rõ cơ chế chia sẻ dữ liệu và bộ nhớ và năng lực tính toán của các CPU.	
L21.	Tìm hiểu và trình bày các thành phần, cơ chế làm việc của công nghệ GPGPU (General-purpose computing on graphics processing units) sử dụng năng lực của các card đồ họa.	

L22.	Tìm hiểu và làm rõ các thành phần của Parallel Computing, làm rõ cơ chế chia sẻ dữ liệu và bộ nhớ và năng lực tính toán của các CPU.	
L23.	Tìm hiểu và trình bày về nguyên lý, cấu tạo, hoạt động của các hệ phân tán (Distributed Systems) (đọc, dịch và báo cáo tóm tắt)	Chapter 17. Distributed Systems
L24.	Tìm hiểu và trình bày về đặc điểm, cấu tạo, các thành phần chức năng của The Linux System (đọc, dịch và báo cáo tóm tắt)	Chapter 18. The Linux System
L25.	Tìm hiểu và trình bày về đặc điểm, cấu tạo, các thành phần chức năng của Windows 7 (đọc, dịch và báo cáo tóm tắt)	Chapter 19. Windows 7
L26.	Tìm hiểu và trình bày về đặc điểm, cấu tạo, các thành phần chức năng của Windows 10	
L27.	Tìm hiểu cơ chế quản lý bảo mật trong hệ điều hành Unix (Linux).	
L28.	Tìm hiểu và trình bày về đặc điểm, cấu tạo, các thành phần chức năng của Windows Server 2016	
L29.	Tìm hiểu và trình bày về cấu tạo, các thành phần chức năng của các hệ điều hành nhúng (Embedded Operating Systems), dẫn dắt ví dụ cụ thể.	
L30.	Tìm hiểu và làm rõ các thành phần của Distributed Computing, làm rõ cơ chế chia sẻ dữ liệu và bộ nhớ và năng lực tính toán của các CPU.	
L31.	Tìm hiểu và trình bày về điện toán đám mây và ứng dụng (Cloud Computing)	
L32.	Tìm hiểu và trình bày về Internet vạn vật (The Internet of Things -IoT)	
L33.	Tìm hiểu và trình bày báo cáo tổng quan về các thế hệ và các dòng họ máy tính cho đến nay, thông số kỹ thuật và năng lực tính toán đi kèm.	
L34.	Tìm hiểu và trình bày báo cáo tổng quan về các thế hệ và các dòng họ hệ điều hành cho đến nay, thông số kỹ thuật và năng lực tính toán đi kèm.	
L35.	Tìm hiểu về trình bày báo cáo tổng quan về công nghệ RAID (Redundant Array Of Independent Disks)	
L36.	Tìm hiểu về cơ chế giao tiếp giữa các chương trình sử dụng Shared Memory và dẫn chương trình minh họa, giải thích chương trình.	

L37.	Tìm hiểu về cơ chế giao tiếp giữa các chương trình sử dụng Pipe và dẫn chương trình minh họa, giải thích chương trình.	
L38.	Tìm hiểu về cơ chế giao tiếp giữa các chương trình sử dụng Message Queue và dẫn chương trình minh họa, giải thích chương trình.	
L39.	Tìm hiểu về cơ chế và môi trường, công cụ phát triển ứng dụng với Universal Windows Platform (UWP), dẫn chương trình minh họa.	
L40.	Tìm hiểu về cơ chế và môi trường, công cụ phát triển ứng dụng trên Windows App Store, dẫn chương trình minh họa.	
L41.	Tìm hiểu về cơ chế và môi trường, công cụ phát triển ứng dụng trên Android App Store, dẫn chương trình minh họa.	
L42.	Tìm hiểu về cơ chế và môi trường, công cụ phát triển ứng dụng trên iOS App Store, dẫn chương trình minh họa.	
L43.	Tìm hiểu và viết báo cáo về DSM (Distributed Shared Memory)	
L44.	Tìm hiểu và trình bày về cơ chế, chức năng quản lý thiết bị ngoại vi trong Windows	
L45.	Tìm hiểu và trình bày về cơ chế, chức năng quản lý thiết bị ngoại vi trong Linux/Unix	
L46.	Tìm hiểu và trình bày về các hàm API làm việc với tiến trình trong HĐH Windows, viết hoặc tìm hiểu chương trình minh họa.	
L47.	Tìm hiểu và trình bày về các hàm API làm việc với files trong HĐH Windows, viết hoặc tìm hiểu chương trình minh họa.	