

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA: Công nghệ thông tin

ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 3, năm học 2023 – 2024

Mã học phần: 233_71ITNW30403

Tên học phần: Hệ điều hành

Mã nhóm lớp học phần: 233_71NWBC30403_01, 233_71NWBC30403_0101

Thời gian làm bài (phút/ngày): 75 phút

Hình thức thi: **Tự luận**

SV được tham khảo tài liệu: Có Không

Giảng viên nộp đề thi, đáp án bao gồm cả **Lần 1 và Lần 2**

Cách thức nộp bài (Giảng viên ghi rõ yêu cầu): Thi trên máy, sinh viên gõ trực tiếp trên khung trả lời của hệ thống thi.

Lưu ý: Sinh viên KHÔNG được sử dụng Tài liệu, mạng Internet và điện thoại di động.

Câu 1 (2 điểm): Tổng quan Hệ điều hành

- Trình bày các chế độ hoạt động của Hệ điều hành
- Trình bày các dịch vụ lõi (core services) của Hệ điều hành nằm ở hạt nhân (kernel)

Câu 2 (2 điểm): Quản lý Tiến trình

- Trình bày các trạng thái của Tiến trình. Vẽ biểu đồ mô tả các trạng thái của các Tiến trình
- Sử dụng hàm system() để viết chương trình C nhằm tạo các tiến trình sau:
 - Tạo thư mục VLU1, VLU2
 - Tạo tập tin TTSV1.txt với nội dung họ tên, mssv, lớp của sinh viên
 - Sao chép TTSV1.txt vào thư mục VLU1, VLU2
 - Nén thư mục VLU2 thành VLU2.tar
 - Xóa thư mục VLU2
 - In nội dung tập tin TTSV1.txt ra màn hình

Câu 3 (2 điểm): Điều phối CPU

- Có 3 tiến trình vào hàng đợi theo thứ tự P1, P2, P3 và có burst time:

Tiến trình	Burst Time (Thời gian thực thi/xử lý)
------------	---------------------------------------

P1	24 ms
P2	8 ms
P3	8 ms

Hãy sử dụng giải thuật điều phối FCFS (first come first served) để vẽ sơ đồ Gantt và tính thời gian chờ, thời gian chờ trung bình, thời hoàn thành trung bình.

- b. Có 4 tiến trình vào hàng đợi theo thứ tự P1, P2, P3, P4 và có burst time:

Tiến trình	Burst Time (Thời gian thực thi/xử lý)
P1	12 ms
P2	6 ms
P3	8 ms
P4	10 ms

Hãy sử dụng giải thuật điều phối RR (Round Robin) với Quantum Time là 4 ms để vẽ sơ đồ Gantt và tính thời gian chờ, thời gian chờ trung bình, thời hoàn thành trung bình.

Câu 4 (2 điểm): Đồng bộ Tiến trình

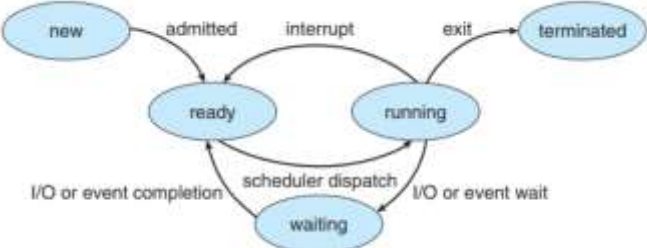
- Trình bày cơ chế hoạt động của PiPe để thực hiện giao tiếp giữa các tiến trình (Inter-Process Communication - IPC).
- Mô tả bài toán đồng bộ hóa Producer – Consumer. Trình bày giải pháp Semaphore để giải quyết bài toán Producer-Consumer.

Câu 5 (2 điểm): Quản lý Bộ nhớ và Quản lý Tập tin hệ thống

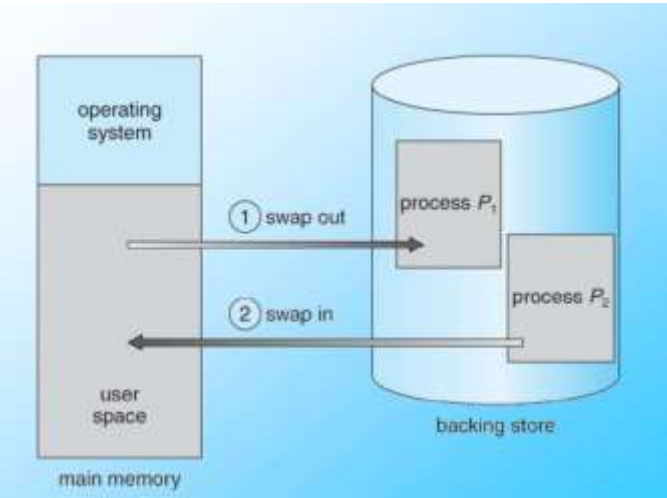
- Trình bày kỹ thuật hoán vị (swapping) để giải quyết phân mảnh bộ nhớ. Vẽ mô hình thao tác của kỹ thuật hoán vị.
- Trình bày 3 loại quyền tập tin cơ bản r, w, x trong hệ điều hành Linux. Các quyền truy cập này được phân chia cho 3 loại người dùng nào ?

ĐÁP ÁP VÀ THANG ĐIỂM

Phần câu hỏi	Nội dung đáp án	Thang điểm	Ghi chú
I. Tự luận			
Câu 1		2.0	
Nội dung a.	<p>- Chế độ hoạt động kép: Để đảm bảo hệ điều hành chạy tốt, phải có khả năng phân biệt giữa việc thực thi mã lệnh của hệ điều hành và việc thực thi mã lệnh của người dùng. Do vậy cần hai chế độ riêng biệt của hoạt động: chế độ người dùng (user mode) và chế độ hạt nhân (kernel mode, còn gọi là chế độ giám sát, chế độ hệ thống, hoặc chế độ đặc quyền).</p> <p>- Timer: Bộ đếm thời gian được dùng để ngăn chặn một chương trình người dùng chạy quá lâu</p>	1.0	
Nội dung b.	<p>Các dịch vụ lỗi nằm ở hạt nhân của hệ điều hành:</p> <ul style="list-style-type: none"> • Thực thi chương trình • Hoạt động I/O • Thao tác hệ thống tập tin • Truyền thông • Phát hiện lỗi • Phân bổ tài nguyên • Kế toán • Bảo vệ an ninh. <p>- 6 nhóm dịch vụ đầu nhằm cung cấp môi trường làm việc thuận tiện cho người dùng.</p> <p>- 3 nhóm sau nhằm đảm bảo sự hoạt động hiệu quả, an toàn của chính hệ thống.</p>	1.0	
Câu 2		2.0	
Nội dung a.	<p>Trạng thái Tiến trình</p> <ul style="list-style-type: none"> • Khi một tiến trình thực thi, nó thay đổi trạng thái. • Trạng thái của một tiến trình được xác định bởi hoạt động hiện tại của nó. • Mỗi tiến trình có thể ở một trong những trạng thái sau: <ul style="list-style-type: none"> ▪ new: Tiến trình vừa được tạo (chạy chương trình) ▪ ready: Tiến trình sẵn sàng để chạy (đang chờ cấp CPU) ▪ running: Tiến trình đang chạy (thi hành lệnh) ▪ waiting: Tiến trình chờ đợi một sự kiện ▪ terminated: Tiến trình kết thúc thi hành lệnh 	1.0	

												
<p>Nội dung b.</p>	<p>Chương trình mẫu:</p> <pre>#include <stdio.h> #include <stdlib.h> int main() { // Tạo thư mục VLU1 và VLU2 system("mkdir VLU1 VLU2"); // Tạo tập tin TTSV1.txt với nội dung họ tên, MSSV, lớp của sinh viên system("echo 'Họ tên: Nguyễn Văn A\\nMSSV: 123456\\nLớp: CNTT1' > TTSV1.txt"); // Sao chép TTSV1.txt vào thư mục VLU1 và VLU2 system("cp TTSV1.txt VLU1/"); system("cp TTSV1.txt VLU2/"); // Nén thư mục VLU2 thành VLU2.tar system("tar -cvf VLU2.tar VLU2"); // Xóa thư mục VLU2 system("rm -rf VLU2"); // In nội dung tập tin TTSV1.txt ra màn hình system("cat TTSV1.txt"); return 0; }</pre>	<p>1.0</p>										
<p>Câu 3</p>		<p>2.0</p>										
<p>Nội dung a.</p>	<table border="1" data-bbox="448 1301 762 1424"> <tr> <td> P1 </td> <td> P2 </td> <td> P3 </td> </tr> <tr> <td>0</td> <td>24</td> <td>32</td> </tr> <tr> <td></td> <td></td> <td>40</td> </tr> </table> <ul style="list-style-type: none"> • Thời gian chờ: <ul style="list-style-type: none"> ▪ P1: 0 ms ▪ P2: 24 ms ▪ P3: 32 ms • Thời gian chờ trung bình: 18.67 ms • Thời gian hoàn thành trung bình: 32 ms 	P1	P2	P3	0	24	32			40	<p>1.0</p>	
P1	P2	P3										
0	24	32										
		40										
<p>Nội dung b.</p>	<p>Tương tự nội dung câu a</p>	<p>1.0</p>										
<p>Câu 4</p>		<p>2.0</p>										
<p>Nội dung a.</p>	<ul style="list-style-type: none"> • Một pipe là một kênh liên lạc trực tiếp giữa hai tiến trình :dữ liệu xuất của tiến trình này được chuyển đến làm dữ liệu nhập cho tiến trình kia dưới dạng một dòng các byte. • Khi một pipe được thiết lập giữa hai tiến trình, một trong chúng sẽ ghi dữ liệu vào pipe và tiến trình kia sẽ đọc dữ liệu từ pipe. Thứ tự dữ liệu truyền qua pipe được bảo toàn theo nguyên tắc FIFO 	<p>1.0</p>										

	<ul style="list-style-type: none"> • Một pipe có kích thước giới hạn (thường là 4096 ký tự) • Một tiến trình chỉ có thể sử dụng một pipe do nó tạo ra hay kế thừa từ tiến trình cha. Hệ điều hành cung cấp các lời gọi hệ thống read/write cho các tiến trình thực hiện thao tác đọc/ghi dữ liệu trong pipe. Hệ điều hành cũng chịu trách nhiệm đồng bộ hóa việc truy xuất pipe trong các tình huống: <ul style="list-style-type: none"> ▪ Tiến trình đọc pipe sẽ bị khóa nếu pipe trống, nó sẽ phải đợi đến khi pipe có dữ liệu để truy xuất. ▪ Tiến trình ghi pipe sẽ bị khóa nếu pipe đầy, nó sẽ phải đợi đến khi pipe có chỗ trống để chứa dữ liệu • Cơ chế này cho phép truyền dữ liệu với cách thức • không cấu trúc. • Ngoài ra, một giới hạn của hình thức liên lạc này là • chỉ cho phép kết nối hai tiến trình có quan hệ chacon, và trên cùng một máy tính. 		
Nội dung b.	<p>Bài toán Producer-Consumer (Nhà sản xuất và Người tiêu dùng) là một bài toán đồng bộ hóa cổ điển trong lập trình hệ thống và quản lý tài nguyên. Bài toán này bao gồm hai loại tiến trình:</p> <ul style="list-style-type: none"> • Producer (Nhà sản xuất): Tạo ra các sản phẩm và đưa chúng vào một bộ đệm (buffer). • Consumer (Người tiêu dùng): Lấy sản phẩm từ bộ đệm để sử dụng. <p>Mục tiêu là đảm bảo sự đồng bộ hóa giữa các nhà sản xuất và người tiêu dùng để tránh các vấn đề như:</p> <ul style="list-style-type: none"> • Underflow: Khi người tiêu dùng cố gắng lấy sản phẩm từ một bộ đệm rỗng. • Overflow: Khi nhà sản xuất cố gắng đưa sản phẩm vào một bộ đệm đã đầy. <p>Bộ đệm thường được thiết kế dưới dạng một hàng đợi với kích thước cố định. Các nhà sản xuất phải chờ nếu bộ đệm đầy, và các người tiêu dùng phải chờ nếu bộ đệm trống.</p> <p>Giải pháp Sử dụng Semaphore: Semaphore là một công cụ cơ bản trong lập trình đồng bộ hóa để điều phối sự truy cập vào tài nguyên chia sẻ. Để giải quyết bài toán Producer-Consumer, ta có thể sử dụng ba semaphore:</p> <ul style="list-style-type: none"> • Semaphore empty: Đếm số lượng ô trống trong bộ đệm (số lượng sản phẩm có thể được đưa vào bộ đệm). Khởi tạo với giá trị bằng kích thước của bộ đệm. 	1.0	

	<ul style="list-style-type: none"> • Semaphore full: Đếm số lượng sản phẩm hiện có trong bộ đệm (số lượng sản phẩm có thể được lấy ra từ bộ đệm). Khởi tạo với giá trị 0. • Semaphore mutex: Đảm bảo chỉ một tiến trình (nhà sản xuất hoặc người tiêu dùng) có thể truy cập vào bộ đệm tại một thời điểm. Khởi tạo với giá trị 1. 		
Câu 5		2.0	
Nội dung a.	<p>Sử dụng kỹ thuật hoán vị (swapping):</p> <ul style="list-style-type: none"> • Chuyển tạm một vài tiến trình đang trong trạng thái blocked hoặc ready ra bộ nhớ phụ, giải phóng bộ nhớ chính để có vùng nhớ trống liên tục đủ lớn cho việc nạp chương trình mới (swap out). • Sau này chương trình bị chuyển tạm ra bộ nhớ phụ sẽ được nạp trở lại vào bộ nhớ chính để tiếp tục thực thi (swap in). • Hệ thống bị chậm lại do thời gian hoán vị một chương trình cũng khá lớn. • Nâng cao mức độ đa chương • Khi tiến trình được mang lại bộ nhớ chính, nó sẽ được nạp vào vùng nhớ: <ul style="list-style-type: none"> ▪ Kết buộc lúc nạp, phải đưa vào vùng nhớ cũ của nó ▪ Kết buộc lúc thi hành, thay đổi thanh ghi tái định vị • Cần sử dụng bộ nhớ phụ đủ lớn để lưu các tiến trình bị khóa. • Thời gian swap chủ yếu là thời gian chuyển tiến trình từ vùng nhớ chính đến bộ nhớ phụ, một tiến trình cần phải có khoảng thời gian xử lý đủ lớn 	1.0	
Nội dung b.	<p>Quyền Truy Cập Tập Tin:</p> <ul style="list-style-type: none"> • r (Read - Đọc): 	1.0	

	<ul style="list-style-type: none"> ▪ Quyền đọc cho phép người dùng xem nội dung của tập tin hoặc liệt kê nội dung của thư mục. ▪ Đối với tập tin: Cho phép đọc nội dung của tập tin. ▪ Đối với thư mục: Cho phép liệt kê các tập tin và thư mục con trong thư mục đó. • w (Write - Ghi): <ul style="list-style-type: none"> ▪ Quyền ghi cho phép người dùng thay đổi nội dung của tập tin hoặc thêm/xóa tập tin trong thư mục. ▪ Đối với tập tin: Cho phép sửa đổi nội dung của tập tin. ▪ Đối với thư mục: Cho phép thêm, xóa, hoặc đổi tên các tập tin trong thư mục đó. • x (Execute - Thực Thi): <ul style="list-style-type: none"> ▪ Quyền thực thi cho phép người dùng thực hiện tập tin (nếu đó là một chương trình hoặc script) hoặc di chuyển vào thư mục. ▪ Đối với tập tin: Cho phép chạy tập tin nếu nó là một chương trình hoặc script. ▪ Đối với thư mục: Cho phép di chuyển vào thư mục và thực hiện các lệnh liên quan đến thư mục. <p>Phân Chia Quyền Truy Cập cho Ba Loại Người Dùng:</p> <ul style="list-style-type: none"> • Người sở hữu (Owner) • Nhóm (Group) • Người dùng khác (Others) 		
	Điểm tổng	10.0	

Ngày biên soạn: 19/07/2024

Giảng viên biên soạn đề thi và đáp án: ThS. Bùi Duy Cường

Ngày kiểm duyệt: 20/07/2024

Trưởng (Phó) Khoa/Bộ môn kiểm duyệt đề thi: ThS. Trần Quang Nhật

Sau khi kiểm duyệt đề thi, **Trưởng (Phó) Khoa/Bộ môn** gửi về Trung tâm Khảo thí qua email: khaothivanlang@gmail.com bao gồm file word và file pdf (được đặt password trên 1 file nén/lần gửi) và nhắn tin password + họ tên GV gửi qua Số điện thoại Thầy Phan Nhật Linh (0918.01.03.09).