***Lớp N7 tênSV+MSV* (điền ngay trước khi làm bài !)**

**Lưu bài làm này với tên tệp là tên Lớp N7 tênSV+MSV.doc.**

**Gởi vào Email** [**htcuoc@gmail.com**](mailto:htcuoc@gmail.com)**. E mail đặt tiêu đề**

**N7-tênSV+MSV. Ai không gửi vào cuối giờ, không nhận bài sau thời điểm đó !**

**Ví dụ: tệp bài làm N7-Nguyễn Văn An-** **D19CQCN0001-B.docx**

**Bài kiểm tra cuối kỳ, môn Hệ Điều Hành.**

**Học phần: Hệ điều hành (Học kỳ 1 2021-2022)**

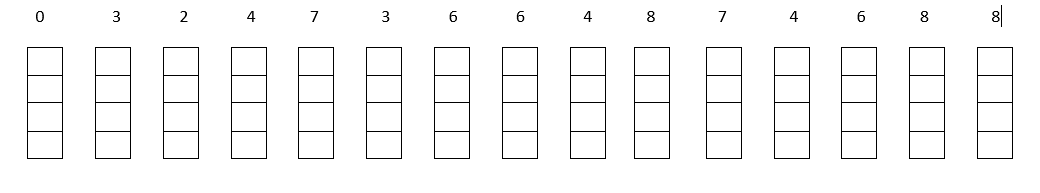
**Thời gian làm bài : 90 phút**

**Đề số 1**

**I. Điểm cho bài kiểm tra học kỳ (3 câu đầu):**

**Câu 1 (3 điểm):**

Một tiến trình được cấp 4 khung để nạp code. Thứ tự các trang nạp theo yêu cầu có trình tự như trong hình. Hãy điền số trang vào các ô khung khi cần nạp trang nếu xẩy ra tình trạng thiếu khung cho trang bằng **phương pháp Tối ưu (OPT)**

****

**Câu 2 (3 điểm):**

1. Trình bày phân chương động bộ nhớ. Phân tích ưu nhược điểm của phương pháp này so với phân chương cố định.
2. Cho ví dụ minh họa cấp phát động với các chiến thuật: **firrst fit, best fit, worst fit**.
3. Khi di chuyển chương sang vị trí khác, cần thay đổi thông tin gì trong khối ánh xạ địa chỉ.

**Câu 3 (4 điểm):**

1. Trình bày khái niệm, tổ chức khi phân trang bộ nhớ.
2. Trình bày thuật ánh xạ địa chỉ khi phân trang bộ nhớ.
3. Không gian bộ nhớ logic của TT gồm có 4 trang, mỗi trang có kích thước 4096 bytes. Bộ nhớ vật lí có 64 khung. Để biểu diễn địa chỉ logic và địa chỉ vật lí cần bao nhiêu bits ?
4. **Cho một địa chỉ logic= 12046. Hãy tính địa chỉ vật lí của địa chỉ logic đó**

**dựa vào bảng trang sau:**

|  |  |
| --- | --- |
| Số trang | Số khung |
| 0 | 13 |
| 1 | 10 |
| 2 | 15 |
| 3 | 5 |

**2. Điểm cho thực hành:**

**Câu 4 (4 điểm): (Điểm tham gia thực hành có mặt 2 buổi: 6 đ và bài tập này cộng lại)**

Cho trước một hàm addcount( ) tính giá trị biến counter, với số vòng lặp chứa trong biến loops. Viết chương trình main() nhận từ bàn phím số đếm cho biến loops. Trong main tạo 2 luồng cùng dùng chung hàm addcount().Chạy thử với các số đếm tăng dần tới khi thấy kết quả sai như dự đoán. Sau đó thực hiện đồng bộ thực thi bằng cách loại trừ lẫn nhau (MUTEX) để đảm bảo kết quả đứng. Sử dụng các hàm: **pthread\_mutex\_lock(&m);, pthread\_mutex\_unlock(&m); pthread\_exit(NULL);**

**(Cho trước các dữ kiện sau: )**

#include <stdlib.h>

#include "common.h"

#include <pthread.h>

pthread\_mutex\_t m; //khai báo đối tượng MUTEX

volatile int counter = 0;

int loops;

void \*addcount(void \*arg)

{

int i;

for (i = 0; i < loops; i++)

{ counter = counter + 1;}

}

**main() { ….. }**

**BÀI LÀM**

**(Làm bài trong tệp này luôn)**