**PROJECT - NGUYÊN LÝ HỆ ĐIỀU HÀNH**

**Học kỳ 1 – 2023-2024**

**Máy ảo sử dụng: Ubuntu VM**



Cách thức nộp bài: 01 folder tên mã số sinh viên (***MSSV***) và họ tên của anh chị ***MSSV\_HọTên*** chứa

- 01 báo cáo mô tả chi tiết cách giải quyết các vấn đề nêu lên trong project (**kể cả hình ảnh minh họa kết quả và các lệnh** anh/chị thực hiện theo đúng trình tự)

- Tất cả code (cả mức kernel và user) có liên quan

*Lưu ý:*

- File báo cáo cần có Họ tên và MSSV và phần tuyên bố như bên dưới

- Code cần có chú thích rõ ràng, đầy đủ

**Tuyên bố: Project này là do chính tôi, *Quách Minh Hớn* (MSSV: B2110078), tự thực hiện, không sao chép của bất kỳ ai. Nếu có bất cứ sao chép nào, tôi hoàn toàn chịu trách nhiệm.**

**PHẦN 3: Cấp phát vùng nhớ**

Anh chị hãy cài đặt 3 thuật toán cấp phát vùng nhớ (First-fit, Best-Fit, Worst-Fit) cho tiến trình, đưa ví dụ cụ thể để test kết quả thực hiện.

# First-fit

#include <stdio.h>

void First\_Fit(int phanVung[], int tienTrinh[], int parSize, int proSize) {

    int i, j, check[proSize], tmp[parSize];

    for (i=0; i<parSize; i++)

        tmp[i] = phanVung[i];

    // gan cac phan tu cua mang phanVung vao mang tmp

    // cap nhat lai mang tmp sau moi lan duyet

    // mang tmp chi dung de printf

    for (i=0; i<proSize; i++) {

        check[i] = 0; // gan gia tri 0 de tim tien trinh nao chua duoc allocated

        printf("%d: ", i+1); // print thu tu cua moi lan duyet

        for (j=0; j<parSize; j++) {

            if (tienTrinh[i] <= phanVung[j]) {

                phanVung[j] -= tienTrinh[i]; // lay phanVung - tienTrinh

                printf("%d --%d--> %d => %d is allocated to partition No.%d", tmp[j], tienTrinh[i], phanVung[j], tienTrinh[i], j+1);

                check[i] = 1; // neu tienTrinh duoc duyet thi gan check = 1

                break;        // va break khoi vong lap

            }

        }

        // neu tienTrinh[i] van bang 0 sau khi duyet qua cac partitions thi thong bao chua duoc allocated

        if (!check[i])

            printf("%d is not allocated", tienTrinh[i]);

        printf("\n");

    }

}

void printPartitionID(int phanVung[], int parSize) {

    int i;

    printf("|-----|-------------|\n");

    printf("| No. |\t Paritions  |\n");

    printf("|-----|-------------|\n");

    for (i=0; i<parSize; i++)

        printf("|  %d  |\t    %d     |\n", i+1, phanVung[i]);

    printf("|-----|-------------|\n");

    printf("\n");

}

void printParAfter(int phanVung[], int parSize) {

    int i;

    printf("\nCac kich thuoc phan vung sau khi allocate:\n");

    for (i=0; i<parSize; i++)

        printf("%d ", phanVung[i]);

}

int main() {

    int phanVung[] = {300, 600, 350, 200, 750, 125};

    int tienTrinh[] = {115, 500, 358, 200, 375, 1300};

    int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size cua mang phanVung

    int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size cua mang tienTrinh

    printPartitionID(phanVung, parSize); // print ID cua cac phan vung

    First\_Fit(phanVung, tienTrinh, parSize, proSize);

    printParAfter(phanVung, parSize);

}

A screenshot of a computer

Description automatically generated

# Best-fit

#include <stdio.h>

#include <limits.h>

int minimum(int bestFit[], int parSize) {

    int i, min=INT\_MAX;

    // gan min = gia tri max cua integer

    for (i=1; i<parSize; i++)

        if (bestFit[i] < min && bestFit[i] != -1)

            min = bestFit[i];

    // tim gia tri nho nhat nhung lon hon -1

    // neu khong tim duoc gia tri nho nhat thi tra ve INT\_MAX

    return min;

}

void Best\_Fit(int phanVung[], int tienTrinh[], int parSize, int proSize) {

    int i, j, parID[6], check[6], tmp[6];

    int bestFit[6], id, min;

    for (j=0; j<parSize; j++)

        tmp[j] = phanVung[j];

    // gan cac phan tu cua mang phanVung vao mang tmp

    // cap nhat lai mang tmp sau moi lan duyet

    // mang tmp chi dung de printf

    for (i=0; i<proSize; i++) {

        check[i] = 0; // gan gia tri 0 de tim tien trinh nao chua duoc allocated

        printf("%d: ", i+1); // print thu tu cua moi lan duyet

        for (j=0; j<parSize; j++)

            bestFit[j] = -1;

        // reset lai mang bestFit

        // gan gia tri -1 vao tat ca cac phan tu cua mang

        for (j=0; j<parSize; j++)

            if (tienTrinh[i] <= phanVung[j])

                bestFit[j] = phanVung[j];

        // gan lai cac gia tri cua phanVung >= tienTrinh vao mang bestFit

        min = minimum(bestFit, parSize);

        for (j=0; j<parSize; j++) {

            if (tienTrinh[i] <= phanVung[j] && phanVung[j] == min) {

                phanVung[j] -= tienTrinh[i];

                printf("%d --%d--> %d => %d is allocated to partition No.%d\n", tmp[j], tienTrinh[i], phanVung[j], tienTrinh[i], j+1);

                check[i] = 1; // neu tienTrinh duoc duyet thi gan check = 1

                break;        // va break khoi vong lap

            }

        }

        if (!check[i])

            printf("%d is not allocated\n", tienTrinh[i]);

    }

}

void printPartitionID(int phanVung[], int parSize) {

    int i;

    printf("|-----|-------------|\n");

    printf("| No. |  Paritions  |\n");

    printf("|-----|-------------|\n");

    for (i=0; i<parSize; i++)

        printf("|  %d  |     %d     |\n", i+1, phanVung[i]);

    printf("|-----|-------------|\n");

    printf("\n");

}

void printParAfter(int phanVung[], int parSize) {

    int i;

    printf("\n\nCac kich thuoc phan vung sau khi allocate:\n");

    for (i=0; i<parSize; i++)

        printf("%d ", phanVung[i]);

    printf("\n");

}

int main() {

    int phanVung[] = {300, 600, 350, 200, 750, 125};

    int tienTrinh[] = {115, 500, 358, 200, 375, 1300};

    int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size cua mang phanVung

    int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size cua mang tienTrinh

    printPartitionID(phanVung, parSize); // print ID cua cac phan vung

    Best\_Fit(phanVung, tienTrinh, parSize, proSize);

    printParAfter(phanVung, parSize); // print kich thuoc cac phan vung sau khi cap phat

}

A screenshot of a computer

Description automatically generated

# Worst-fit

#include <stdio.h>

#include <limits.h>

int maximum(int bestFit[], int parSize) {

    int i, max=INT\_MIN;

    // gan max = gia tri max cua integer

    for (i=1; i<parSize; i++)

        if (bestFit[i] > max && bestFit[i] != -1)

            max = bestFit[i];

    // tim gia tri nho nhat nhung lon hon -1

    // neu khong tim duoc gia tri nho nhat thi tra ve INT\_MAX

    return max;

}

void Best\_Fit(int phanVung[], int tienTrinh[], int parSize, int proSize) {

    int i, j, parID[6], check[6], tmp[6];

    int bestFit[6], id, max;

    for (j=0; j<parSize; j++)

        tmp[j] = phanVung[j];

    // gan cac phan tu cua mang phanVung vao mang tmp

    // cap nhat lai mang tmp sau moi lan duyet

    // mang tmp chi dung de printf

    for (i=0; i<proSize; i++) {

        check[i] = 0; // gan gia tri 0 de tim tien trinh nao chua duoc allocated

        printf("%d: ", i+1); // print thu tu cua moi lan duyet

        for (j=0; j<parSize; j++)

            bestFit[j] = -1;

        // reset lai mang bestFit

        // gan gia tri -1 vao tat ca cac phan tu cua mang

        for (j=0; j<parSize; j++)

            if (tienTrinh[i] <= phanVung[j])

                bestFit[j] = phanVung[j];

        // gan lai cac gia tri cua phanVung >= tienTrinh vao mang bestFit

        max = maximum(bestFit, parSize);

        for (j=0; j<parSize; j++) {

            if (tienTrinh[i] <= phanVung[j] && phanVung[j] == max) {

                phanVung[j] -= tienTrinh[i];

                printf("%d --%d--> %d => %d is allocated to partition No.%d\n", tmp[j], tienTrinh[i], phanVung[j], tienTrinh[i], j+1);

                check[i] = 1; // neu tienTrinh duoc duyet thi gan check = 1

                break;        // va break khoi vong lap

            }

        }

        if (!check[i])

            printf("%d is not allocated\n", tienTrinh[i]);

    }

}

void printPartitionID(int phanVung[], int parSize) {

    int i;

    printf("|-----|-------------|\n");

    printf("| No. |  Paritions  |\n");

    printf("|-----|-------------|\n");

    for (i=0; i<parSize; i++)

        printf("|  %d  |     %d     |\n", i+1, phanVung[i]);

    printf("|-----|-------------|\n");

    printf("\n");

}

void printParAfter(int phanVung[], int parSize) {

    int i;

    printf("\n\nCac kich thuoc phan vung sau khi allocate:\n");

    for (i=0; i<parSize; i++)

        printf("%d ", phanVung[i]);

    printf("\n");

}

int main() {

    int phanVung[] = {300, 600, 350, 200, 750, 125};

    int tienTrinh[] = {115, 500, 358, 200, 375, 1300};

    int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size cua mang phanVung

    int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size cua mang tienTrinh

    printPartitionID(phanVung, parSize); // print ID cua cac phan vung

    Best\_Fit(phanVung, tienTrinh, parSize, proSize);

    printParAfter(phanVung, parSize); // print kich thuoc cac phan vung sau khi cap phat

}

A screenshot of a computer

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HẾT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*