**Итоговая контрольная работа по курсу «Архитектура компьютеров»**

**Вариант 649**

*Дайте краткий ответ на следующие задания.*

1. **В чём заключается процесс обучения нейронной сети?**

*Различают алгоритмы обучения с учителем и без учителя.*

*Процесс обучения с учителем представляет собой предъявление сети выборки обучающих примеров. Каждый образец подается на входы сети, затем проходит обработку внутри структуры НС, вычисляется выходной сигнал сети, который сравнивается с соответствующим значением целевого вектора, представляющего собой требуемый выход сети.*

*Алгоритмом обучения без учителя мы называем такие алгоритмы, где нет явных указания для алгоритма как ему поступать, а есть только общая оценка всех его действий в процессе решения задачи.*

1. **Дайте определение термину «главный процесс» (master) в контексте программирования на MPI.**

*Главный процесс занимается вводом/выводом данных, и их распределением между остальными узлами. Все остальные процессы ожидают от главного часть массива, вычисляют сумму элементов и передают результат назад.*

1. **Приведите статические способы минимизации конфликтов по управлению**.

*Число тактов, теряемых при приостановках из-за условных переходов, может быть уменьшено двумя способами:*

*1. Обнаружением является ли условный переход выполняемым или невыполняемым на более ранних ступенях конвейера.*

*2. Использование слотов задержки, во время получения результата перехода вычислять другие команды.*

1. **Что такое «кубит»? Чем он отличается от обычного бита?**

***Кубит*** *- квантовый разряд, или наименьший элемент для хранения информации в квантовом компьютере. Фундаментальное отличие кубита заключается в том, что он, в отличие от бита, может находиться в состоянии квантовой суперпозиции.*

*Дайте развёрнутый ответ на следующие задания.*

1. **Выделите критерии и сравните по ним кэши первого и третьего уровня.**

*Кэш-память первого уровня или L1– очень маленькая, но самая быстрая и наиболее важная микросхема памяти. Ни в одном процессоре ее объем не превышает нескольких десятков килобайт. Работает она без каких-либо задержек. В ней содержатся данные, которые чаще всего используются процессором. Количество микросхем памяти L1 в процессоре, как правило, равно количеству его ядер. Каждое ядро имеет доступ только к своей микросхеме L1. Кэш-память третьего уровня (L3) – еще более объемная, но и более медленная схема памяти. Тем не менее, она значительно быстрее оперативной памяти. Ее размер может достигать нескольких десятков мегабайт. В отличие от L1 и L2, она является общей для всех ядер процессора.*

1. **Какие проблемы архитектуры VLIW решает архитектура EPIC?**

*Эта архитектура направлена на то, чтобы упростить аппаратное обеспечение и, в то же время, извлечь как можно больше «скрытого параллелизма» на уровне команд, используя большую ширину «выдачи» команд (WIW -Wide Issue-Width) и длинные (глубокие) конвейеры с большой задержкой (DPL — Deep Pipeline-Latency), чем это можно сделать при реализации VLIW или суперскалярных стратегий. EPIC упрощает два ключевых момента, реализуемых во время выполнения. Во-первых, его принципы позволяют во время исполнения отказаться от проверки зависимостей между операциями, которые компилятор уже объявил как независимые. Во-вторых, данная архитектура позволяет отказаться от сложной логики внеочередного исполнения операций, полагаясь на порядок выдачи команд, определенный компилятором. Более того, EPIC совершенствует возможность компилятора статически генерировать планы выполнения за счет поддержки разного рода перемещений кода во время компиляции, которые были бы некорректными в последовательной архитектуре.*

1. **Какие задачи эффективно решаются векторными процессорами? За счёт чего это достигается?**

*В задачах моделирования реальных процессов и объектов, для которых характерна обработка больших массивов чисел в форме с плавающей запятой, массивы представляются матрицами и векторами, а алгоритмы их обработки описываются в терминах матричных операций. Как известно, основные матричные операции сводятся к однотипным действиям над парами элементов исходных матриц, которые, чаще всего, можно производить параллельно. Задачи на распознавание образов, ассоциативную и оптическую обработку данных, вычисление наибольшего правдоподобия, обработку сигналов и решение систем дифференциальных уравнений.*

1. **Разверните следующий цикл по три итерации:**

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

double x = a + i \* h;

y[i] = f(x);

}

Ответ:

int i;

for (i = 0; i < N - 2; i += 3) {

double x1 = a + i \* h;

double x2 = a + (i + 1) \* h;

double x3 = a + (i + 2) \* h;

y[i] = f(x1);

y[i + 1] = f(x2);

y[i + 2] = f(x3);

}

for (; i < N; i++) {

double x = a + i \* h;

y[i] = f(x);

}

*Дайте ответ на следующие задания и поясните его.*

**9.** **Дан следующий псевдокод:**

for (int i = 0; i < N; ++i)

for (int j = 0; j < M; ++j)

a[i + j] += a[i + j] \* b[i + j];

Младший программист хочет эффективно распараллелить вычисления в цикле при помощи OpenMP. Возможно ли это?

Ответ: Нет, так как итерации внутреннего цикла зависят от внешнего цикла. Если выполнение любого связанного цикла изменяет любое из значений (в нашем случае a[i + j]), используемых для вычисления любого из счетчиков итераций, то поведение не определено.

Например, при i=0; j= 5 значение a[i + j] будет одно, а при i=2; j= 3 мы будем использовать предыдущее значение. В итоге, при распараллеливании вычислений мы можем использовать неправильное значение и поведение будет не определено.

**10. Дан следующий псевдокод:**

int f(int x, int y, int z) {

if (x < 0) return -1;

if (y < 0) return -1;

if (z < 0) return -1;

if (x + y + z == 0) return -1;

return (int)sqrt(x \* x + y \* y + z \* z);

}

Какие методы смогут минимизировать конфликты по управлению при выполнении данного кода на конвейерном процессоре. Поясните свой ответ.

1. Пока результат перехода задерживается, вычисляем другие команды.
2. Использовать слоты задержки
3. Использовать всегда ложное условие