PR2

Muhammad Irfan Amrullah / 1706039585

Kelompok Soal Pertama: Process Memory

1. Program merupakan rangkaian algoritma Panjang yang berupa file eksekusi,, berisikan intruksi-intruksi untuk memecahkan suatu masalah.

Proses merupakan tugas yang dijalankan oleh computer, disimpan di penyimpanan utama dan akan hilang ketika computer direboot atau dimatikan.

Thread, atau kadang-kadang disebut sebagai proses ringan (lightweight) merupakan unit dasar dari utilisasi CPU yang didalamnya terdapat ID thread, program counter, register, dan stack.

1. Contoh Program:

* Notepad.exe
* Safari.exe`
* Counter Strike.exe

Contoh Process:

* Printer program yang memonitor tinta
* Daemon
* Syslog

Kelompok Sola Kedua: Process Management

1. a. - ps -A : Display every active process on a linux system in generic

* ps -x : Select all processes owned by you
* ps -C sshd : select specific process by its name, this will also display all its child processes

b. User dapat melihat bagian STARTED ketika mengecek dengan ps aux

c. D – Sleep yang gabisa diganggu gugat

R – sedang dalam queue run

S – Sleep yang bias di ganggu

T – Berhenti, entah oleh karena memang diperintahkan ato karena sedang di trace

Z – Process zombie, sudah dimatikan tapi tidak ditutup oleh parent process yang membuatnya

W – gapunya resident pages

< -- process sangat penting

N – proses kurang penting

L – proses tersebut memiliki halaman yang langsung terkunci ke memori

d. PID – nomer id proses

USER – username dari yang punya proses

PR – prioritas dari proses

NI – nice value

VIRT – total memory virtual yang dipakai task

RSS – Real memory usage

S – kode status proses

%MEM – Berapa banyak memory yang dipake proses

%CPU – Berapa banyak CPU yang dipake proses

TIME+ -- total penggunaan CPU

COMMAND – nama proses termasuk argument jika ada

1. itu akan membuat proses baru yang mana akan menjadi child process dari yang memanggil. Setelah child process baru dibuat, kedua process akan menjalankan perintah berikutnya setelah fork() system call. Oleh karena itu kita harus membedakan parent dari child, ini bias dilakukan dengan mengetest value fork()
2. getppid(): mengembalikan id proses orang tua yang melakukan proses calling

getpid(): mengembalikan Id proses dari yang melakukan proses calling

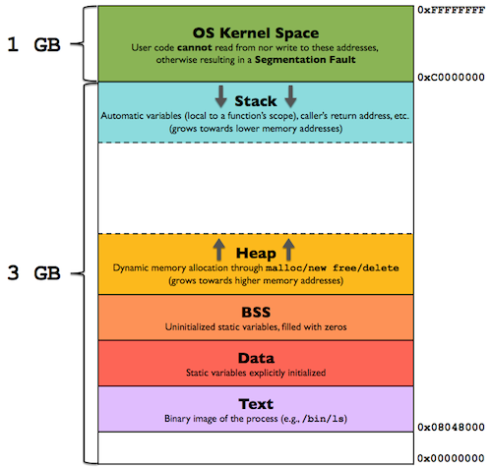
1. Zombie process: sebuah process yang punya eksekusi komplit tapi belom di bunuh oleh parent processnya. Sebagai hasilnya dia masih memiliki process entry di process table dan PID.

Orphaned process: sebuah proses yang berjalan padahal parent processnya sudah selesai atau dimatikan

Daemon process: sebuah orphaned process yang disengaja agar bisa berjalan di background process.

1. Ada banyak cara untuk mencegah terjadinya zombie process, salah satunya adalah menggunakan wait() system call. Ketika parent process memanggil wait(), setelah pembentukan sebuah child, ini mengindikasikan bahwa dia akan menunggu child untuk komplit dan akan memastikan exit status dari sebuah child. Apabila sudah terlanjur terbentuk ia harus di wait() oleh parentnya, jadi membunuh parentnya akan efektip untuk membuh zombie.

Kelompok Soal Ketiga: Memory

1. 

* Text: bagian text, juga dikena sebagai code segment. Berisi excutable instructions dari sebuah program.
* Data: kependekan dari “Initialized data segment”. Bagian ini mengandung virtual address dari sebuah program yang berisi variable global dan static yang di mulai oleh seorang programmer.
* BSS: BSS segment juga dapat dikatakan “unintialized data segment”. Data the segment ini di initialized oleh OS kernel ke arithmetic 0 sebelum program dieksekusi.
* Stack: area stack mengandung program stack, contohnya struktur LIFO biasanya ditaruh di memory addresses yang lebih tinggi tepat dibawah OS kernel space.
* Heap: segment dimana alokasi memori dinamik ditaruh, contohnya untuk mengalokasi memory yang diminta programmer untuk variable yang mana sizenya baru dapat diketahui setelah dijalankan dan tidak bisa ditentukan secara static oleh compiler sebelum program dieksekusi.

1. Karena virtual memory membuat kita bisa menjalankan lebih banyak aplikasi di system ketimbang apabila kita Cuma menggunakan physical memori saja. Mengapa physical memory saja tidak cukup karena system itu sendiri saja sudah menggunakan physical ram yang ada. Apabila kita terus membuka aplikasi, suatu saat pasti ada situasi dimana kita tidak bisa membuat aplikasi lagi karena kehabisan physical ram. Disini virtual memory bekerja.

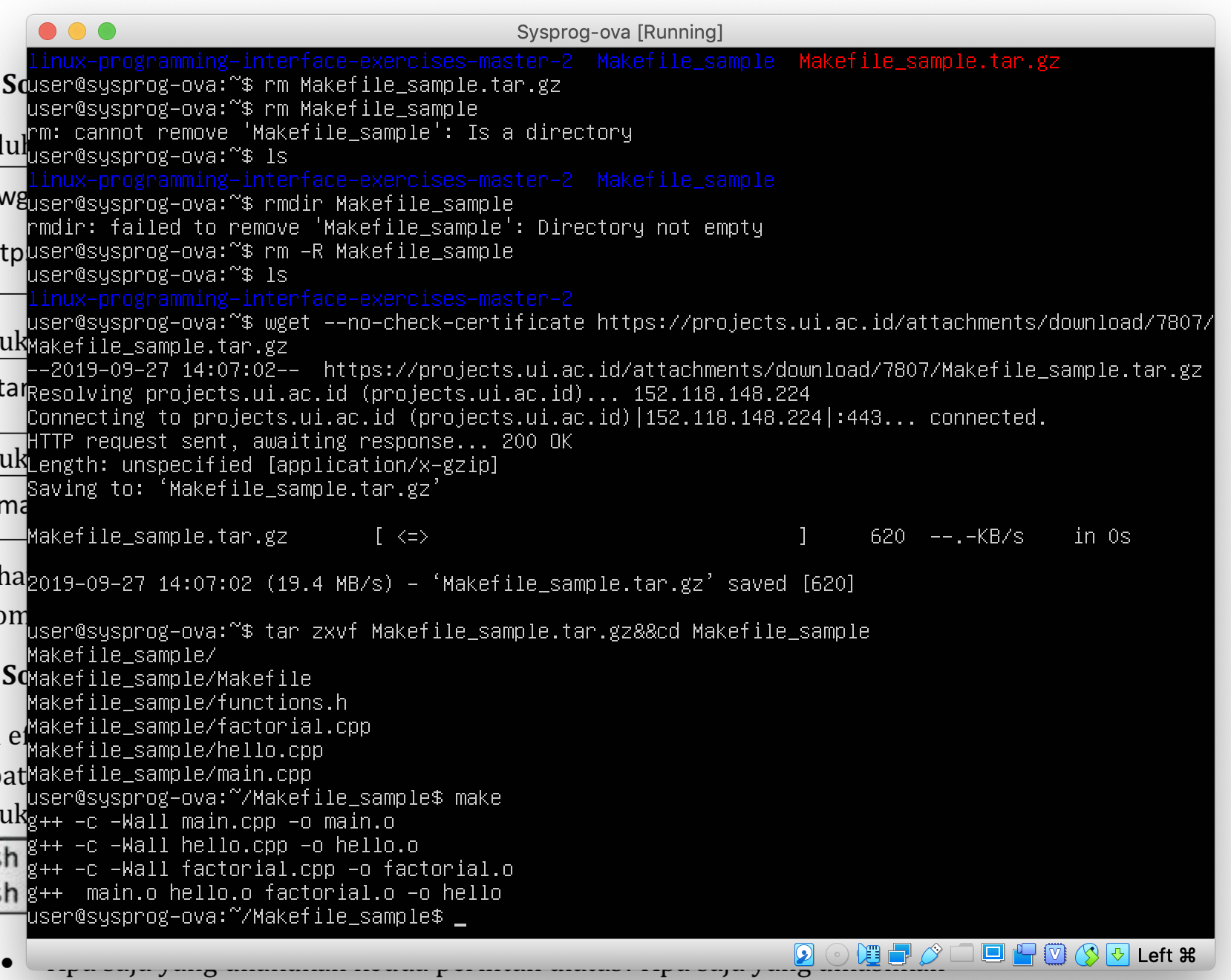
Advantages:

* Kita bisa menjalankan lebih banyak aplikasi bersamaan
* Kita bisa menjalankan aplikasi yang besar dengan real RAM yang sedikit
* Kita ga harus beli memory RAM lagi

Disadvantages:

* Aplikasinya jadi lebih lambat
* Jadi butuh waktu yang lebih banyak untuk pindah antar aplikasi
* Hard drive menjadi lebih sedikit

1. Memory leak terjadi ketika programmer membuat sebuah memory di heap dan lupa menghapusnya. Memory leaks merupakan hal yang serius untuk program seperti daemons dan server yang sesuai definisinya tidak akan pernah selesai. Untuk menghindari memory leak, memory ditaruh di heap harus selalu dibebaskan ketika tidak digunakan lagi.

Kelompok Soal Keempat: Makefile

1. Main.cpp ketika decompile menjadi main.o

Hello.cpp ketika decompile menjadi hello.o

Factorial.cpp di compile menjadi factorial.o

Main.o, hello.o, factorial.o decompile menjadi hello

Tahapan kompilasi:

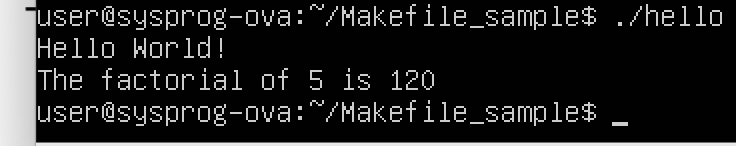
G++ -c -Wall main.cpp -o main.o = compile main.cpp menjadi file dengan nama main.o

G++ -c -Wall hello.cpp -o hello.o = compile hello.cpp menjadi file dengan nama hello.o

G++ -c -Wall factorial.cpp -o factorial.o = compile factorial.cpp menjadi file dengan nama factorial.cpp

G++ main.o hello.o factorial.o -o hello = link kode objek file main.o, hello.o dan factorial.o menjadi sebuah file executable dengan nama hello

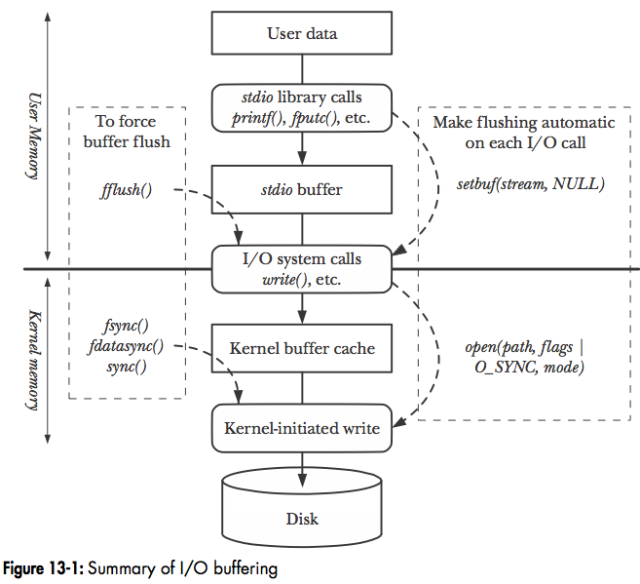
Output file hello:

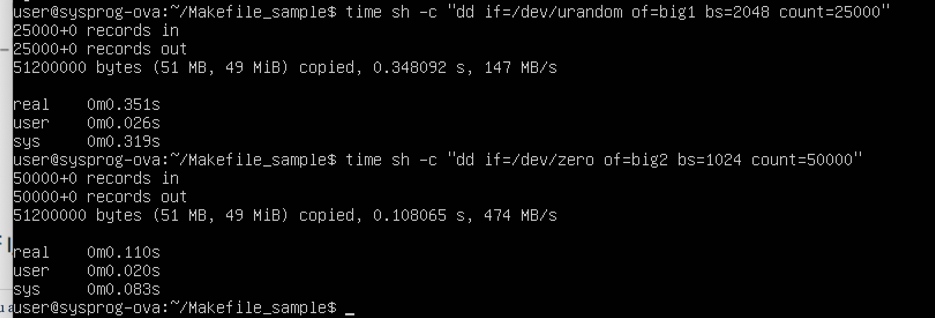


Kelompok Soal Kelima: I/O Buffering

1. I/O itu sangat lambat, untuk bagian output apabila kamu memiliki buffer, kamu membolehkan CPU untuk mengirim info ke device dan purapura bahwa itu sudah selesai sehingga dia bisa langsung menjalankan hal berikutnya tanpa membuang-buang waktu diam saja untuk memberikan informasi dengan kecepatan yang seharusnya.

Dari perspektif input, sinyal ga selalu sampai ketika CPU sudah siap untuk menerimanya, jadi sinyal harus ditaro di tempat lain sehinggal dia bisa dihandle oleh CPU. Buffer akan memastikan bahwa sinyal input yang dating ga ada yang hilang ketika CPU sedang menghandle sinyal yang akan datang.

1. 

* Fungisnya adalah menentukan waktu eksekusi sebuah operasi spesifik, dalam hal ini adalah waktu yang dibutuhkan untuk menulis sebuah file

Perbedaannya adalah di block size dan file content

Persamaannya adalah file sizenya (51200000 bytes)

Kelompok soal keenam: process Identifier

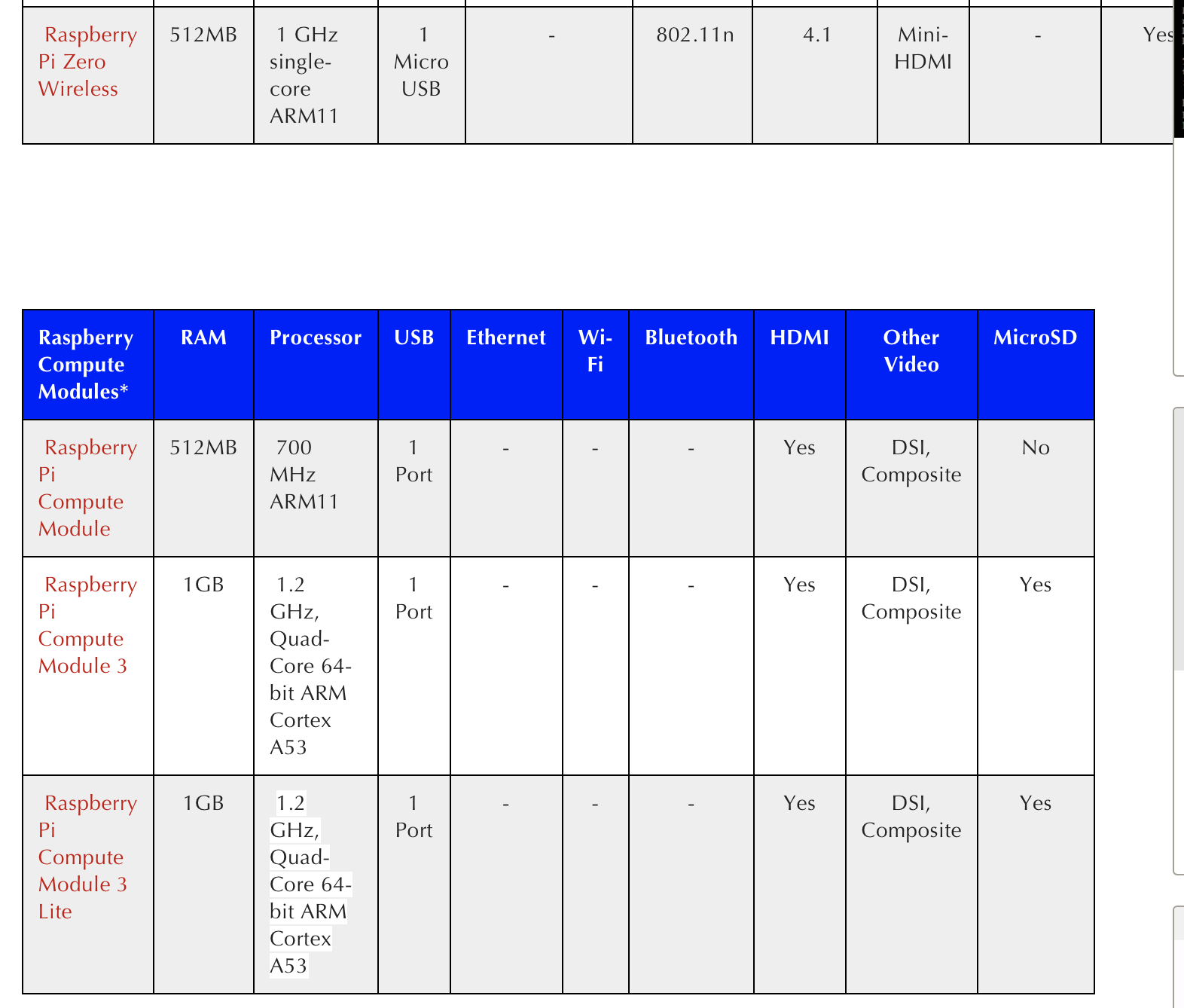
1. Swapper or sched punya id 0 dan bertanngung jawab untuk paging, bahkan sebenernya merupakan bagian dari kernel ketimbang dari process normal user-mode. Process ID 1 biasanya merupakan init process yang tugas utamanya bertanggung jawab untuk memulai dan mematikan system.

Sebenarnya process id 1 bukan special di reserved untuk init process, dia memiliki id 1 karena secara natural yang dinyalakan pertama oleh kernel.

PID 0 adalah parent dari PID 1. PID 1 adalah parent dari semua process.

1. Ada, yaitu 32768 untuk 32-bit dan 2^22 untuk 64 bit. Apabila ada pid yang lebih besar dari maksimal maka akan tidak dialokasikan.
2. Iya dapat diubah, dengan maksimal 2^22 atau sekitar 4194304 di mesin 32 bit. Alasannya adalah sebenernya tidak ada alasan khusus tentang ini, memang teknis dari sananya seperti itu

Kelompok soal ketujuh: Raspberri Pi



1. Noobs, RISC OS, Linutop, Raspbian, OSMC, OpenELEC, Windows IoT Core, Lakka, RaspBSD, Retropie

Raspberry pi mendukung beberapa OS dan sering sekali tidak memiliki satu pun os ketika baru dibeli. Tapi biasanya dia dikirim dengan sd card yang telah diisi oleh NOOBS. OS yang didalamnya terdapat banyak OS yang tinggal kamu pilih mau pakai yang mana.

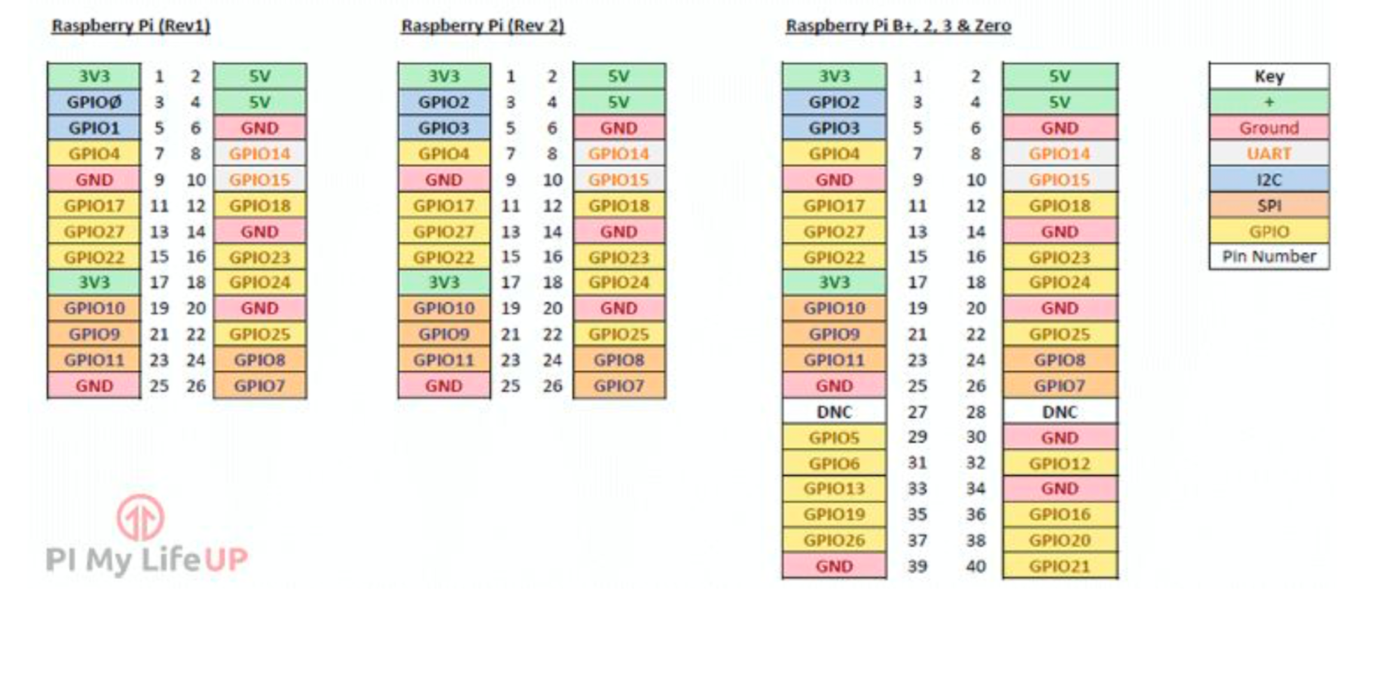
1. SBC:

* Ada memory dan storage
* Punya colokan usb
* Udah lengkap, tinggal pakai
* Bisa secara penuh jalanin OS

Microcontroller:

* Memorinya terbatas, biasanya kurang dari 1MB
* Storange ada diflash, kecil.
* Support untuk USB baru tidaka da, biasanya support usb 2.0 bahkan 1.1

1. - PWM (pulse-width modulation) sebuah Teknik untuk mengatur power. Kita pakai itu untuk mengatur jumlah power yang masuk ke motor dan seberapa cepat dia berputar. Hardware PWM tersedia di GPIO12, GPIO13, GPIO18, GPIO19

* SPI(Serial Peripheral Interface) pin dapat digunakan untuk dihubungkan ke SPI devices. Kurang lebih sama speerti I2C tapi menggunakan protocol yang berbeda.
* I2C (Inter Integrated circuit) pins membolehkanmu untuk terhubung dan berbicara ke modul hardware yang mendukung protocol ini. Biasanya mengambil 2 pin.
* UART(Universal Asynchronous Receiver/Transmitter) serial pin yang digunakan untuk berbicara dengan device lain
* GND (Ground) pin yang kamu guenakan untuk meng-ground device kamu. Ga masalah yang mana yang dipakai asal mereka terhubung ke line yang sama
* DNC(Do not Connect)
* Power pins mengambil power langsung dari raspberry pi
* GPIO adalah standar pin yang bisa dipakai untuk menyalakan dan mematikan device
* Bentuk pin: