

공장자동화를 위한 컴퓨터 구조

박 규 호

(한국과학기술원 전기 및 전자공학과의교수)

1. 서 론

생산성 향상과 이에 따르는 국제 경쟁력 강화 등 공장자동화는 선진국으로의 도약에 필수적인 과정이라 할 수 있으나 이같은 공장자동화에는 여러 분야의 기술이 집대성된 여러 시스템 기술이 요구되며 해결되어야 하는 여러 문제점이 있다.

한편, 급격히 발전되어온 반도체 기술에 의한 저가격 대용량 기억소자의 출현, 값싼 마이크로 프로세서의 등장 및 컴퓨터 기술의 발달은 이와 같은 여러 문제점을 해결할 수 있는 수단을 제공하고 있다. 공장자동화는 종래의 단순하고 국부적인 자동화에서 복잡다양하고 광범위한 전체 공장 자동화로 발전되어가고 있다.

자동화 공정에서의 대상 시스템은 규모면에서 보면 모터와 같은 소형 시스템에서부터 정유, 제철, 전력 시스템 등과 같은 대형 시스템이 있으며 내용면에서는 생산산업체(manufacturing industry)와 처리산업체(processing industry)로 크게 구분할 수 있고 소형 시스템은 마이크로 프로세서를 포함한 실시간 처리 컴퓨터를 사용하여 구성된다. 그러나 대형 시스템에서는 기존의 마이크로 프로세서 응용기술만으로는 제어 시스템의 성능에 관한 여러가지의 요구 조건을 해결할 수 없게 되었다.

자동화용 컴퓨터 기술은 마이크로 프로세서 응용기술 뿐만 아니라 분산 처리, 데이터 베이스, 다중 프로세서

등의 응용까지 포함해야 대형 시스템 구성을 실현시킬 수 있다.

즉, 최근의 공장 자동화는 대량생산 위주에서 다품종 소량생산으로 바뀌고 있다. 앞으로의 생산 제품의 대부분은 중소형 기업에서 불규칙하고 다양한 소비자의 요구에 따라서 생산되어야 하므로 이에 적합한 생산 시스템이 필요하다. 이같은 목적으로 컴퓨터는 공장의 자동화에 있어서 각종의 생산도구들을 융통성있게 운용하는데 있어서 가장 중요한 요소이다.

일반적으로 부품들이 대량생산될 때는 고정된 자동화 설비를 이용하는 것이 경제적으로 유리하나, 소량 생산인 경우는 이와같은 설비를 가설하는 비용에 비하여 비경제적이다.

융통성을 갖는 자동화 시스템은 두가지 방식을 절충하는 것으로서 제품의 확일성을 증가시킬 수 있을 뿐만 아니라, 수요에 따라서 여러가지의 스타일을 선택할 수 있는 다양성까지 갖는다.

이같은 목적으로 소프트웨어의 대처로서 기존 line의 큰 변화없이 다품종 소량생산에 대처할 수 있는 FMS (flexible manufacturing system)가 필요하다.

FMS는 자동화된 작업의 흐름으로 연관되는 workstation들의 그룹과 이들을 연관시키고, 각 workstation에서 작업을 하는 로봇, 그리고 부품의 제조 및 조립을 관리하는 종합제어 컴퓨터로 구성된다.

자동화 시스템을 이루기 위해서는 다음 4가지 요소들

이 갖추어져야 하며 시스템 구성을 위해서 반드시 고려해야 한다.

- (1) 통신
- (2) 부재한 제어
- (3) 실시간 처리
- (4) 사람과 기계와의 접속장치

앞으로의 제조 시스템, 자동화된 job shops, 그리고 산업 로봇들에서의 컴퓨터 하드웨어는 분산된 센서들과 actuator들이 부착된 다중 프로세서들이 서로 상호연관을 갖는 형태로 구성된다. 이같은 처리능력의 분산은 신뢰도의 증가와 실시간 응답이 빨라지는 반면에 소프트웨어 설계 및 구성의 문제가 아직도 많이 남아 있다.

본 고에서는 우선 공장자동화를 위한 컴퓨터의 기본적인 특질인 실시간성에 대하여 정의하기로 하고 그 다음으로 컴퓨터 구조에 대하여 논하기로 한다.

2. 실시간 컴퓨터의 정의

실시간 컴퓨터는 흔히 embedded 컴퓨터라고도 지칭하며 컴퓨터가 한 시스템의 component로 사용되는 것으로부터 명명된 것이다. 따라서 일반적으로 입력 데이터의 수집, 응답시간 등이 주어진 제약 조건내에 행하여지는 특성을 가진 컴퓨터가 요구되고 이를 실시간 컴퓨터(real time computer - RTC)라 칭한다.

2.1 RTC의 정의

RTC를 일반적으로 말하면 입력을 감지(sense)하여 그 데이터에 대한 의도하는 operation을 수행하고 빠른 시간내에 응답하는 컴퓨터를 일컫는다. 이와같은 정의는 대단히 애매하므로 입력(input), 응답 시간(response time), 제어(control)의 의미를 좀 더 명확히 할 필요가 있다.

입력: 분류하면 외부제어에 의한 입력(externally controlled input)과 내부제어에 의한 입력(internally controlled input)으로 대별되며 전자는 컴퓨터와 독립적으로 발생하는 입력신호, 예를 들면 지진신호와 같은 것이다. 내부제어에 의한 입력신호의 가장 좋은 예는 magnetic tape reader에 서의 데이터 입력 등이다. RTC에서 input은 외부제어에 의한 입력을 의미한다.

응답 시간: 데이터 입력 시점으로부터 제어를 위한 출력신호가 나오기까지의 시간을 의미하며 주어진 응답시간내에서 출력이 나와야 한다.

제어: 제어에는 직접제어와 간접제어로 분류되며 직접 제어란 closed loop control을 의미하고 indirect control은 human intervention에 의한 제어를 뜻한다.

위와 같은 용어들의 정의로부터 RTC에서 실시간 처리란 외부제어 입력에 대하여 주어진 응답시간 내에서 일어나는 데이터 처리나 직접제어를 수행하는 것을 의미한다. 이와 상대적인 용어는 interactive processing으로서 외부제어 입력 및 reproducible data를 포함하여 간접제어를 수행하는 처리를 뜻한다.

2.2 실시간 시스템의 시간 특성⁴⁾

실시간 시스템이 특성을 나타내는 parameter 들은 다음과 같다.

- i) 최대수행시간(maximum execution time - MET)
각 process 또는 device들의 최대 수행시간
- ii) 최소 CPU 요청간격(minimum request interval - MRI)
각 process 또는 device가 CPU를 요청하는 최소한의 시간 간격

이때 MET/MRI를 load factor라 부르며 single processor 시스템의 경우 load factor가 1 이하이어야 모든 process를 service해 줄 수 있다. 이 조건을 'over run free'라 한다. RTC의 설계 요건은 over run free를 만족시키는 것이다.

실시간 컴퓨터에서 요구되는 MET 및 MRI는 그 경우에 따라 다르지만 그림 2.1, 2.2와 같다.

3. 실시간 컴퓨터의 형태

전형적인 실시간 컴퓨터 시스템의 형태는 다음과 같이 분류된다.⁷⁾

- 1) host-peripheral 시스템

전형적으로 쓰이는 시스템으로서 그림 3.1과 같다. 이 시스템의 경우 host computer는 입출력 기능, numerical computation, information management 및 user interface를 모두 담당하여야 한다.

- 2) parallel processor 첨가 시스템

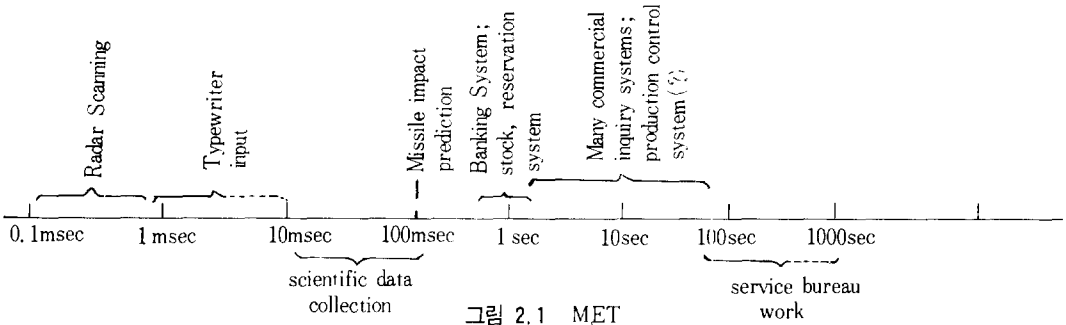


그림 2.1 MET

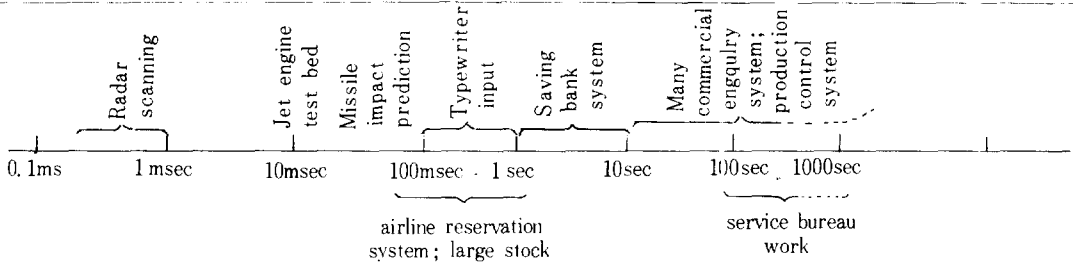


그림 2.2 MRI

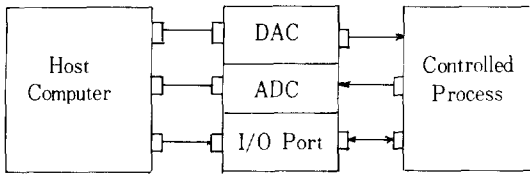


그림 3.1 Typical host-peripheral system

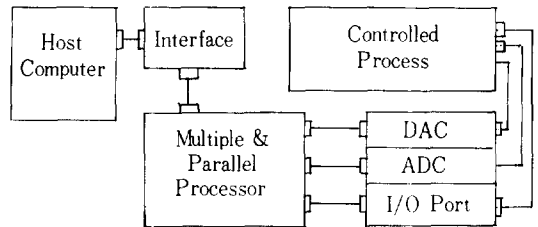


그림 3.3 I/O processor

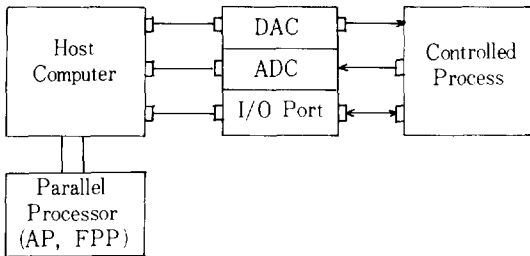


그림 3.2 Parallel processors

host-peripheral에서 CPU의 load를 경감시키는 자연스러운 방법으로서 array processor 또는 floating point processor를 사용하는 것이다. 그 형태는 그림 3.2와 같다. 이렇게 함으로서 numerical computation에 대한 부담은 감소되나 입출력 등의 부담은 그대로 존속한다.

3) I/O processor 부가 시스템

이 시스템의 기본 목적은 실시간 process와 그렇지

않은 일반 process (non-real time process)를 분리하여 host processor 시스템은 모든 non-critical real time process를 수행한다. I/O processor 시스템 쪽은 다중 processor 및 intelligent interface로 구성되어 외부 device들과 연결된다. 그 형태는 그림 3.3과 같다.

4) hierarchical multiple processors

이 시스템은 그림 3.4에 도시한 바와 같이 master-slave로 분리되어 각각의 역할을 가지고 시스템이 운용되는 형태이다. 맨 상위에 있는 processor는 바로 다음 level의 processor를 제어하고 이 processor들은 그 다음 level의 processor를 제어하는 형태를 갖는다. 가장 상위에 있는 processor는 central database를 제공하며 자주 사용되지 않는 복잡한 transaction을 수행하며 이들은 빠른 응답시간을 요구하지 않는다. 상위 processor에서 하위로 내려갈수록 실시간성을 요구한

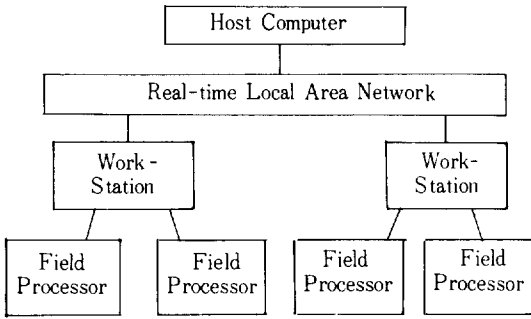


그림 3.4 Hierarchical multiple processors

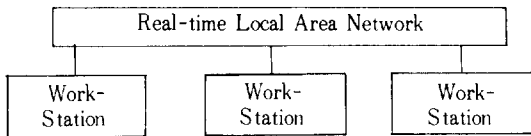


그림 3.5 Horizontal distributed processors

신량이 거의 동등하게 분포되어 있을 때 바람직하다. 이는 그림 3.5에 도시하였다.

4. 공장자동화용 컴퓨터 시스템의 예^{1)~3)}

최근 일본이나 미국 등지에서 공장자동화를 위한 컴퓨터 시스템 개발이 활발히 진행되고 있으나 그 구조 및 운영에 대한 표준을 정하지 못하고 있다. 본 절에서는 현재 미국 NBS (national bureau of standard) 와 일본의 히다치 회사에서 연구 개발되고 있는 공장자동화를 위한 컴퓨터 시스템을 위주로 하여 다루고자 한다.

앞으로의 제조 시스템, 자동화된 job/shop, 그리고 산업 로봇들을 제어하기 위한 컴퓨터구조는 여러가지 센서들과 actuator들이 부착된 프로세서들이 서로 상호 연관된 형태로 분산된 형태를 갖는다. 이같은 처리능력의 분산은 신뢰도의 증가와 실시간 응답이 빨라지는 반면에 소프트웨어 설계 및 구성의 문제가 많다.

생산 시스템에서의 제어 계층은 일반적으로 5개의 주요한 수준으로 나눌 수 있다. 각 수준은 몇가지의 제어를 가지며, 또 여러가지의 단계나 모듈들로 나누어진다.

다음 그림에서 제어 기능은 사각형으로 나타내고, 점선은 상부 계층에서 하부 계층의 작업 흐름을 나타낸다. 각 기능 상자는 내부 제어를 위한 자신의 제어기들을 갖는다. 모든 기능들은 실선으로 나타나 있는 통신 라인과 각 숫자로 표시된 통신 node를 통하여 정보를 교환한다.

다.

이와같은 구조는 실시간 컴퓨터의 구조로서 대단히 적합하다. 그 이유는 실제 실시간 응용예를 보면 대형 컴퓨터의 성능을 요구하는 일 보다는 단순한 다수의 task로 이루어진 것이 대부분이기 때문이다.

5) horizontally distributed processors

이 구조의 경우 processor들은 logical하게 또한 physical하게 거의 동등하다고 볼 수 있다. Horizontal system의 주 특징은 분산제어와 point-to-point 통신이라 할 수 있다. 특히 이 구조는 각 processor들이 다른 processor들과 통신을 하는 확률이 높고 또한 이통

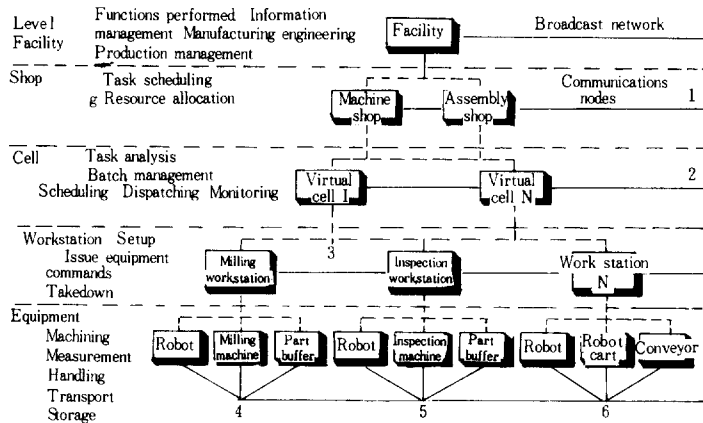


그림 4.1 자동화 생산 시스템의 계층적 구조

1) facility

제어 계층의 가장 높은 수준은 생산공학, 정보관리 및 생산관리의 3가지 주요한 부분 시스템으로 이루어진다. 생산공학은 부분 기기들의 설계를 위한 CAD 및 생산공정의 계획등을 다룬다. 정보관리는 가격 및 투자 평가, 소비자 요구등에 대한 정보들을 관리한다. 생산관리는 중요한 project를 계획하고, 장기간의 스케줄 구성 및 생산자원 요구사항 등을 다룬다. 이 수준에서 만들어진 계획은 다음 단계인 shop 제어 시스템으로 넘어간다.

2) shop

이 수준에서는 두가지의 주요한 모듈들을 통하여 shop floor 상에 있는 작업들과 task 및 자원관리를 한다. Task 관리는 작업 순서의 스케줄, 장비의 유지와 같은 일을 한다. 자원관리는 cell 수준의 제어시스템과 생산작업자들에게 작업장, 작업을 위한 장비, 그에 필요한 재료들을 할당한다.

3) cell

이 수준에서의 제어기는 비슷한 부분들의 batch 작업들을 순서화 한다. Cell은 동적인 생산 제어구조에서 workstation 처리 시스템의 시분할을 허용할 경우에 가상적인 cell들의 모임으로 볼 수 있다. Cell내의 모듈들은 시스템 task의 성능을 제어하고, 자원의 가용도

를 분석하며, 작업의 진행도를 보고하고, 작업들의 스케줄 설정, 자원 준비, task 감시등을 담당한다.

4) workstation

이 수준은 shop floor 상의 장비들에 대한 그룹을 설정하고, 서로간의 일을 조정한다. 전형적인 workstation은 로봇, 기기장비, 재료창고, 그리고 제어컴퓨터로 구성된다.

5) equipment

이들 제어기들은 shop floor 상에 있는 로봇이나 수치 제어 기계등과 직접 관련된다. 이들 시스템들은 재료보관, 운송, 재료처리등의 기본적인 수준의 기능을 담당한다.

앞에서 설명한 계층적 구조 중 workstation과 equipment level을 그 목표로 하며 workstation과 workstation은 LAN으로 연결되어 그 전체가 하나의 distributed real time computer를 형성하도록 한 구성예를 다음 그림에 보였다.

특히 real time environment에는 상황에 따라 여러개의 event가 불규칙적으로, 동시에 일어나게 된다. 그러므로 이러한 조건에서 여러 개의 real time task들을 동시에 수행하기 위하여 여러개의 동시에 수행되는 process들을 여러개의 computer에서 수행시키는 distributed real time feature를 필요로 한다.

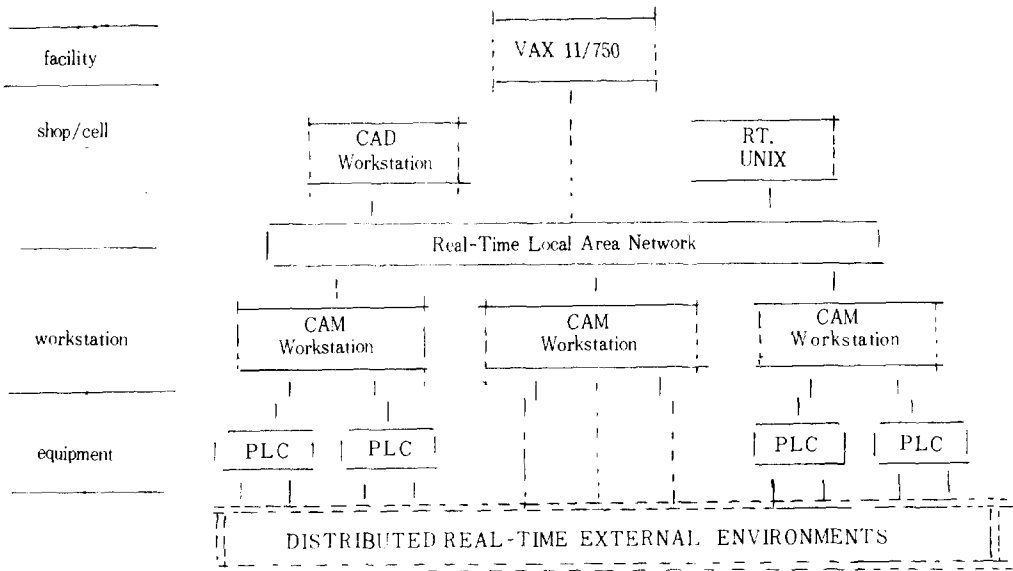


그림 4.2 분산 실시간 컴퓨터 구성 예

즉, 여러개의 process가 동시에 수행되기 위해서는 서로 관계있는 process들간의 정보 교환을 위한 IPC (interprocess communication) 기능이 필요하며 여러 real time process들을 효과적으로 제어하기 위한 제어 process도 필요하다. 이러한 목적으로 distributed real time kernel은 distributed feature를 갖고 design되어야 한다.

그림 4.2에 도시한 분산처리의 장점은 다음과 같다.

① computation이 매우 복잡한 경우 이를 보다 쉬운 여러개의 문제로 나누어서 여러 process들에게 분산시켜 처리하게 한다. 이렇게 함으로써 memory와 시간의 절약을 가져오게 된다. 또한 각 process는 보다 전문적인 일에 종사할 수 있게 되며 이는 각 process에 대한 program이 좀더 쉬워짐을 의미한다.

② 전체 system을 여러개의 process에 분산시킴으로써 system의 확장성을 높게 된다. 즉 system이 점점 커지는 경우 전체 system을 다 바꾸지 않고 지역적인 부분만을 변경함으로써 시스템 확장이나 축소가 가능하다.

③ 여러개의 task들에 대한 처리를 하나의 중심적인 process가 처리하게 된다면 그러한 task들에 의해 처리를 하기 위해서는 process가 매우 고속으로 운용되어야만 한다.

④ 어떠한 real time task를 수행하는 process가 어떠한 이유로 인해 더 이상 일을 수행하지 못하게 되는 경우 다른 process가 대신 일을 수행하게 할 수 있으므로 전체 system의 신뢰성이 높아지게 된다.

6) 통신망구조

앞에 설명한 시스템의 구조 중 통신망 구조에 대한 논의가 필요하다.

공장의 자동화와 같은 시스템은 제어 대상이 지역적으로 분산되어 있으므로 국지 통신망이 많이 사용된다. 하드웨어 가격이 낮아지고 통신 기술의 발전으로 최근에는 분산 시스템으로 실시간을 요구하는 응용분야에서 여러가지의 요구사항들을 만족시켜 나간다.

국지 통신망의 구조는 여러가지가 있으나 공장 자동화와 같은 응용 분야에서는 신뢰도, 응답시간들이 중요한 요소이므로 사무 자동화를 위한 통신망에 비하여 선택범위가 제한된다.

일반적으로 버스 형태와 ring 형태가 널리 사용될 수 있다. 두가지의 특징은 서로 상반되는 면이 있다. 버스

형태는 하나의 케이블에 여러가지의 기능을 갖는 컴퓨터들이 접속장치를 통하여 연결된다. 이 형태의 특징은 접속장치가 수동 소자로 구성되므로 구성이 간단하며 신뢰도가 높다.

통신망을 사용하기 위한 프로토콜은 CSMA(carrier sense multiple access)방식이 주로 사용되고 있다. CSMA방식은 통신망 사용권이 불규칙하게 결정되므로 응답시간을 결정하기 힘들다.

버스 형태에서 다른 프로토콜은 token-passing 방식으로서, 최대 지연 시간을 보장할 수 있으나 신뢰성에서 CSMA 방식에 비하여 낮다.

통신망 사용 프로토콜은 에러의 영향과 긴급 메시지 송수신, 응답 시간 제약 등을 고려하여 결정해야 한다.

Ring 형태는 여러개의 컴퓨터들이 ring의 모양으로 연결된다. 이 형태는 전송로에서 데이터의 충돌이 없으므로 통신망 사용에 대한 지연시간을 보장할 수 있다. 또한 잡음의 영향을 받지 않고 대역폭이 넓은 광섬유를 사용할 수 있는 장점이 있다. 그러나 통신망 접속장치가 능동소자로 구성되어 사고에 따라서는 통신망 전체에 영향을 줄 수 있으므로 신뢰도가 낮은 단점이 있다.

5. 시스템 소프트웨어

계층구조에서 제어 수준이 통신망에 연결된 여러 컴퓨터에 분산되고, 개별 시스템은 각 수준에서 작은 여러가지의 task들로 수행된다. 대부분 제어 프로그램은 서로 통신하는 다른 프로그램이 어디에 있는지 알 필요가 없어야 한다. 그렇게하면 두개의 제어 프로세서 사이의 데이터 교환은 그들의 위치에는 서로가 무관해질 수 있다. 하나의 프로세스 간의 통신을 위한 기능으로 가능하다. 프로그램은 이름에 의해 통신 경로를 열면 통신 관리 프로그램에서 이름으로부터 실제 프로세스의 주소로 바꾸어 적절한 통신 방법을 찾으면 된다. 이 같이 하여 제어 프로그램은 서로 격리되어 있어도 단지 표준화된 통신 방식으로 메시지 교환을 할 수 있다. 외부 장치에 직접 연결되지 않은 제어 모듈은 다른 프로세서로 옮겨질 수 있다.

이같은 환경을 제공하기 위해서는 통신망과 이를 기반으로 분산 시스템을 운용할 수 있는 분산 운영 시스템이 필요하다. 시간 제약을 받는 여러 프로세스를 효율적으로 수행시킬 수 있는 프로세스 관리 기능과 분산

시스템에서의 메시지 교환을 원활히 할 수 있는 통신기능이 중요한 위치를 차지한다. 일반적인 운영체제는 컴퓨터가 갖는 자원들을 효율적으로 관리하는 데에 역점을 두는 반면에 공장 자동화와 같은 분야에서는 실시간 처리가 가능하도록 하며, 시스템 신뢰도 향상에 주안점을 둔다.

실시간 서비스는 deadline 내에 서비스가 제공되지 못했을 경우에 야기되는 문제의 크기에 따라서 실시간 작업수와 우선순위를 정하여 제공한다. 일반적으로 심각한 문제를 야기시킬 경우, 심각하지는 않으나 deadline 이내에 수행되어야 할 경우, 응답 시간과 관련이 없는 작업으로 나누어 우선순위를 정한다. 심각한 문제를 야기시킨 작업이 있는 경우 응답 시간을 확신할 수 있는 범위 내에서 작업량을 결정해야 한다.

일반적으로 신뢰성을 높이는 문제와 빠른 응답 제공에는 서로 상반된 일이 필요하다. 신뢰성을 높이기 위해서는 복잡한 운용 시스템이 되기 쉽고, 빠른 응답을 위해서는 시스템의 단순화가 필요하다. 그러므로 실시간 시스템에서 신뢰성을 높이기 위한 시간적인 손실과 요구되는 응답 시간은 응용 분야에 따라서 적절한 수준에서 결정해야 한다.

분산 시스템에서 각 제어 프로그램과 이들을 운용하는 통신, 데이터 관리, 프로그램 관리 시스템들 간의 통신을 위하여 공통된 언어가 필요하다. 통신 시스템에서의 언어는 가상적인 통신 경로의 이름 설정, 사용권, 사용 방식 등과 같은 기능에 대한 것들이다. 프로세스 관리 언어는 분산 시스템 전반에 걸쳐서 요구되는 프로세스 관리의 공통된 syntax 들이다.

6. 사용자 인터페이스

공장 자동화를 위한 분산처리 시스템의 유지 및 관리를 위하여 오퍼레이터, 프로그래머, 관리인들이 시스템과의 접속장치를 가져야 한다. 간단한 그래픽스와 메뉴 방식, 그리고 자연어와 비슷한 고급 언어가 공장자동화 시스템을 조작하는 숙련되지 않은 사용자에게 필요하다. 인간의 특성을 연구한 결과에 의하면, 사람은 한번에 단지 7가지 정도의 상이한 정보를 인식할 수 있으며 그림에 의한 정보가 문자나 테이블보다 쉽게 이해된다.

미국의 NBS (national bureau standard)에서는 도형에 의한 심볼들을 시스템안에 있는 대상들과 시스템의

상태를 나타내기 위해 개발하고 있다. 이들 심볼들은 사용자가 배우기 쉽고 사용하기 간편하며 쉽게 시스템을 인식할 수 있어야 한다. 이같은 수단은 사용자들이 제어 시스템에게 도식적으로 새로운 task 들을 어떻게 수행할 것인가를 가르칠 수도 있다.

7. 결론

컴퓨터가 공장자동화에 있어서 중요한 위치를 차지하고 있다. 현재 공장 자동화에 있어서 난점은 개발 가격이 높고, 시스템 구조가 부족한 점도 요인이 된다.

공장자동화를 위한 컴퓨터 시스템 구성에서 문제점은 하드웨어와 통신망에서 보다는 소프트웨어에 있다. 현재 고기능을 갖는 마이크로 프로세서, 외부 접속장치 등은 공장자동화에서 요구되는 컴퓨터를 쉽게 구성할 수 있다.

소프트웨어에서는 아직도 해결되지 않은 문제들이 많다. 병행처리가 가능하도록 하며 신뢰도와 성능에 따라서 재구성이 가능한 분산 운용시스템이 개발되어야 한다. 이같은 특성을 갖는 운용 시스템은 기존의 시스템과 다른 개념에서 고려되어야 한다. 기본적으로 고려되어야 할 사항들은 신뢰도, 융통성, 보장된 응답들이다.

공장자동화를 위한 컴퓨터 구성을 위한 제어 특성과 이에 적합한 구성 요소들을 고찰하여 보았다.

공장자동화가 널리 보급되기 위해서는 컴퓨터와 생산 시스템 구성 요소들이 종합적으로 연구 개발되어야 하며, 표준화된 기능 및 접속방식이 이루어져야 한다.

참 고 문 헌

- 1) C. McLean, M. Mitchell and E. Barkmeyer, "A Computer Architecture for Small-Batch Manufacturing," IEEE Spectrum, pp. 59-64, May, 1983.
- 2) Hiromu Tanimoto, "Factory Automation: An Automatic Assembly Line for the Manufacture of Printers," IEEE Computer, vol. 17, no. 12, Dec., 1984.
- 3) N. Komoda, K. Kera & T. Kubo, "An Autonomous, Decentralized Control System for Factory Automation," IEEE Computer, Dec., 1984.
- 4) Mellichamp, Real-Time Computing, Van Nostrand Reinhold, 1983.

- 5) 김명환, 최은호, “실시간 처리 컴퓨터 시스템,” 전기
학회지 제33권 제9호 1984년 9월.
- 6) James D. Schoeffler, “Distributed Computer Sys-
tems for Industrial Process Control,” IEEE
Trans. Computer, Feb., 1984.
- 7) Karl Johan Astrom, “Process Control-Past,
Present, and Future,” IEEE Control Systems
Mag. Aug., 1985.